ORIGINAL

#14 AF/2751

RAB 97-002



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



In re Application of: ) Art Unit: *2751*
*Belgard, R.* )
) Examiner: *Nguyen, T*
Serial No.: *08/905,356* )
)
Filed: *8/4/97 as continuation of parent application*)
*serial no. 08/458,479 filed 6/2/95* )
)
For: *Fast Address Generator For Reducing* )
*Virtual-Linear-Physical Address Conversion Time* )
---------------------------------------------------- )

TRANSMITTAL OF APPEAL BRIEF

Assistant Commissioner
of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

Enclosed herein in triplicate is Appellant's Brief in the above application in support of the Notice of Appeal filed January 24, 2000. A fee in the amount of $150 (small entity) is enclosed. In the event an extension of time is required, please consider this transmittal as a petition therefor. For any such fees, or other deficiencies, please charge deposit account no. 501-244.

Respectfully submitted,

Law

Date: March 20, 2000

J. Nicholas Gross, Attorney, Reg. No. 34,175

*I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Commissioner of Patents and Trademarks, this 20th day of March 2000.*

ORIGINAL

OIPE JCG
MAR 22 2000
PATENT & TRADEMARK OFFICE

RAB 97-002

RECEIVED
MAR 24 2000
GROUP 2700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF APPEALS AND INTERFERENCES

| | |
|---|---|
| In re Application of: *Belgard, R.* | ) Art Unit: *2751* |
| | ) Examiner: *Nguyen, T* |
| Serial No.: *08/905,356* | ) |
| Filed: *8/4/97 as continuation of parent application serial no. 08/458,479 filed 6/2/95* | ) |
| For: *Fast Address Generator For Reducing Virtual-Linear-Physical Address Conversion Time* | ) |

#14 Appeal Brief 3/28/00

APPELLANT'S BRIEF UNDER 37 C.F.R. 1.192

**J. Nicholas Gross**
Attorney for Appellant

1385 Mission Street
Suite 240
San Francisco, CA 94103
415-551-8298

*I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Commissioner of Patents and Trademarks, this 20th day of March 2000.*

03/24/2000 STEFERRA 00000089 08905356
01 FC:220 150.00 OP

## TABLE OF CONTENTS

LIST OF EXHIBITS

| | |
|---|---|
| A | Copy of application serial no. 08/905,356 |
| B | Copy of preliminary amendment filed August 4, 1997 |
| C | First Office Action issued on October 16, 1998 |
| D | Crawford (U.S. Patent No. 5,321,836) |
| E | Toy (U.S. Patent No. 4,400,774) |
| F | Amendment filed November 24, 1998 |
| G | Second Office Action issued February 1, 1999 |
| H | Amendment and response dated July 30, 1999 from Applicant |
| I | Third (and Final) Office Action mailed 10/26/99 by PTO |

## TABLE OF AUTHORITIES

INTRODUCTION AND REVIEW OF PROSECUTION HISTORY

This brief is presented in support of the Notice of Appeal filed for application serial no. 08/905,356; a copy of the disclosure is enclosed as Exhibit A. The present application is a continuation of U.S. application serial no. 08/458,479 (parent) filed on June 2, 1995, now U.S. Patent No. 5,895,503. A preliminary amendment filed in the present case is attached as Exhibit B.

A first Office Action in the present application was issued on October 16, 1998 and is attached hereto as Exhibit C. As a first grounds, the Examiner rejected claims 1-5 for double patenting in light of the parent application, and claims 38, 39, 41 and 42 under §112 for overbreadth, on the grounds that they purportedly were directed to a single means system. *See e.g.*, page 2 of Exhibit C. The Examiner also contended that claims 1-5 were obvious under § 103 in light of reference to Crawford (U.S. Patent No. 5,321,836) – a reference cited by the Applicant. Claims 38 – 81 were rejected as obvious in light of Crawford taken in combination with Toy (U.S. Patent No. 4,400,774) – also cited by the Applicant. The Examiner acknowledged that Toy was directed to a non-segmented system (see page 5) but argued that the cache address unit 125 in Toy was a "speculative address generator" because it used prior address bits to locate cached physical addresses. The Crawford and Toy references are attached hereto as Exhibits D and E respectively.

In response, Applicant canceled claims 1 – 5 to overcome the double patenting rejection, and amended independent claim 38 to overcome the § 112 rejection in an amendment filed November 24, 1998, attached hereto as Exhibit F. Furthermore, new claims 82 – 112 were added for consideration. The Applicant explained that the Toy reference was not even performing a virtual - to linear address conversion, and therefore it could not contain any "linear" addresses of the type set out in the claims, and that there was no suggestion anywhere in Toy or Crawford that would lead one of skill in the art to combine their teachings.

The Examiner then issued a second Office Action on February 1, 1999, in which he withdrew the rejections under § 112 for claims 38, 39, 41 and 42. He further withdrew the aforementioned § 103 rejections, but then issued a *new* rejection of claims 38-81 and new claims 82 – 113 under § 102, contending at that point that the Toy reference now *anticipated* the claims. Apparently in response to the amendment of the claims pertaining to "linear" addresses introduced by the Applicant, the Examiner argued now that the address translation

buffer 102 of Toy "...*inherently* generates a linear address and physical address based on the virtual address, which comprises segment, page, word address information...." (emphasis added). A copy of this Office Action is enclosed as Exhibit G.

In a response dated July 30, 1999, Applicant again amended the independent claims (38, 43, 49, 54, 57, 61, 66, 70, 74, 77, 82, 86, 89, 95, 101 and 107) to make the distinctions over Toy more apparent to the Examiner. A copy of this Amendment is enclosed as Exhibit H. The Applicant further pointed out the Examiner seemed to be changing his position on the teaching of the Toy reference between the first and second office action, in that the Examiner was now arguing that the Toy reference included linear addresses simply because it performed virtual to physical address translations. Applicant then gave a detailed accounting of the Toy reference, including a thorough demonstration of how it could not possibly be using linear addresses of the type disclosed in the invention. So as to make the distinction even more clear, however, Applicant pointed out that not all of the virtual address is even translated in Toy (see e.g., column 3, ll. 62 – 68) and, on that basis, Applicant amended the claims to indicate that the linear address in the inventions is a result of *all* of the virtual address being translated, not simply a portion.

In a third (and now Final) Office Action mailed 10/26/99, the Examiner changed course again, and now rejected the claims under § 103 in light of Crawford and Toy for substantially the same reasons as he had done in the first office action.[1] This correspondence from the PTO is attached as Exhibit I.

Because the claims have now been finally rejected Applicant filed a Notice of Appeal on January 24, 2000, which was within the statutory period of time for doing so.

[1] The third Office Action indicates that a "new search" and "new art" were used to reject the claims in light of the Applicant's amendments (see page 2, third paragraph), but it can be seen quite clearly from comparing the rejections of the first Office Action and the third Office action that the Examiner in fact used exactly the same art, and exactly the same arguments, with little consideration given to the substance of the intervening amendments made by the Applicant.

1. REAL PARTY IN INTEREST

The real party in interest in this case is the applicant, Richard A. Belgard.

2. OTHER RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to appellant or appellant's representative that could directly or be directly affected by or have a bearing on the Board's decision in the instant case.

3. STATUS OF CLAIMS

Claims 38 – 112 are pending.

4. STATUS OF AMENDMENTS

No other amendments have been filed subsequent to the Examiner's office action of October 26, 1999.

5. SUMMARY OF THE INVENTION

A. General Background information – Virtual/Linear/Physical Address and Segmentation/Paging concepts

Computer system address spaces are typically separated into two distinct categories: (1) virtual address spaces used by system software; and (2) physical address spaces used by system hardware. This separation allows programmers to think in terms of their conceptual models, and to design computer software programs without reference to specific hardware implementations. During the actual execution of programs by the computer system, however, these virtual addresses must be reconciled by translating software program virtual addresses into actual physical addresses that can be accessed in a computer memory subsystem.

To accomplish this translation, it is known to use various operations classified generally in the art as segmentation only, paging only, and a combination of segmentation and paging. There are now briefly described.

In a segmentation portion of an address translation system, the address space of a user program (or programs cooperatively operating as processes or tasks), is regarded as a

collection of segments which have common high-level properties, such as code, data, stack, etc. The segmented address space is referenced by a 2-tuple, known as a virtual address, consisting of the following fields: <<s>,<d>>, where <s> refers to a segment number (also called identifier or locator), and <d> refers to a displacement or offset, such as a byte displacement or offset, within the segment identified by the segment number. The virtual address <17,421>, for example, refers to the 421st byte in segment 17.

The end result is that the segmentation portion of the address translation mechanism, using information created by the operating system of the computer system, translates the virtual address into a linear address in a linear address space. In some systems which use segmentation alone, or in which paging is not enabled, this linear address is used directly the hardware to perform a memory access.

In a paging portion of an address translation system, a linear (or intermediate) address space can be thought to consist of a group of pages. Each page is the same size (*i.e.* it contains the same number of addresses in the linear space). The linear address space is mapped onto a multiple of these pages, commonly, by considering the linear address space as the 2-tuple consisting of the following fields: <<page number >,< page offset>>. The page number (or page frame number) determines which linear page is referenced. The page offset is the offset or displacement, typically a byte offset, within the selected page.

In a system using paging, the real (physical) memory of a computer is also conceptually divided into a number of page frames, each page frame capable of holding a single page. Individual pages in the real memory are then located by the address translation mechanism during a paging operation by using one or more page tables created for, and maintained by, the operating system. These page tables are typically caches, and correspond to a mapping from a page number to a physical page frame. This physical page frame is combined with other portions of the linear address to generate a physical address that is used to perform a conventional memory access.

B. Memory access systems that use combined segmentation and paging

Address translation mechanisms which employ both segmentation and paging are well known in the art. There are two common subcategories within this area of virtual address translation schemes: address translation in which paging is an integral part of the

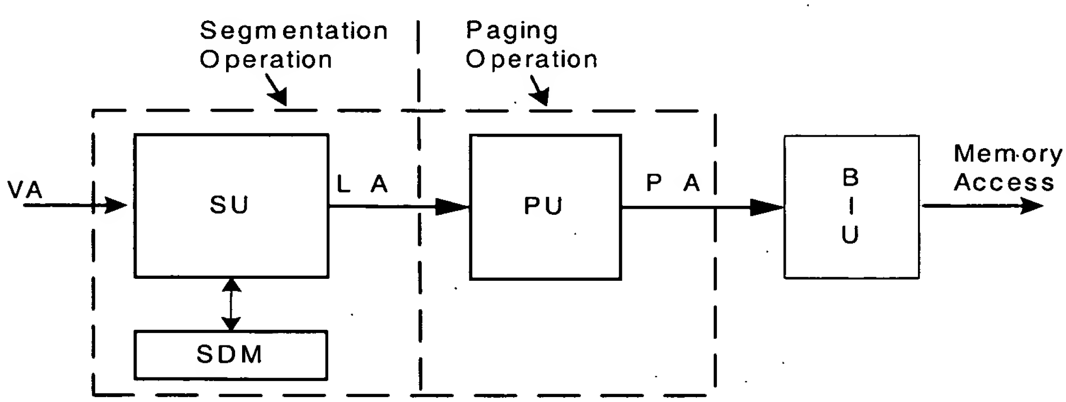
segmentation mechanism; and, address translation in which paging is independent from segmentation.

In prior art address translation mechanisms where paging is an integral part of the segmentation mechanism, the page translation can proceed in parallel with the segment translation since segments must start at page boundaries and are fixed at an integer number of pages. The segment number typically identifies a specific page table and the segment offset identifies a page number (through the page table) and an offset within that page. While this mechanism has the advantage of speed (since the steps can proceed in parallel) it is not flexible (each segment must start at a fixed page boundary) and is not optimal from a space perspective (*e.g.* an integer number of pages must be used, even when the segment may only spill over to a fraction of another page). An example of an "integrated" segmentation/paging system is the Toy reference identified by the Applicant during the prosecution of the present case, and which is now cited by the Examiner against the present claims. Other typical examples of such prior art in which paging is integral to segmentation is the Multics virtual memory, developed by Honeywell and described by the book, "The Multics System", by Elliott Organick.

In prior art address translation mechanisms where paging is independent from segmentation, page translation generally *cannot* proceed until an intermediate, or linear, address is first calculated by the segmentation mechanism. As is apparent, there is no linear address calculation required or performed in so-called "integrated" segmentation/paging systems discussed immediately above. In the independent segmentation/paging systems, the resultant linear address is then mapped onto a specific page number and an offset within the page by the paging mechanism. The page number identifies a page frame through a page table, and the offset identifies the offset within that page. In such mechanisms, multiple segments can be allocated into a single page, a single segment can comprise multiple pages, or a combination of the above, since segments are allowed to start on any byte boundary, and have any byte length. Thus, in these systems, while there is flexibility in terms of the segment/page relationship, this flexibility comes at a cost of decreased address translation speed. An example of an "independent" segmentation/paging system is the Crawford reference identified by the Applicant during the prosecution of the present case, and which is now cited by the Examiner against the present claims.

C. Brief Explanation Of Some Key Aspects Of Applicant's Invention

Accordingly, a key limitation of the above prior art methods and implementations of the second type described above (i.e., where segmentation is independent from paging) is that the linear address must be fully calculated by the segmentation mechanism each time before the page translation can take place for each new virtual address. Only subsequent to the linear address calculation, can page translation take place. Thus, to perform a memory access based on a virtual address, two distinct operations are always required: virtual to linear translation; and then linear to physical translation. In high performance computer systems computer systems, this typically takes two full or more machine cycles and is performed on each memory reference. This additional overhead often can reduce the overall performance of the system significantly. An example of this type of operation is shown in the Crawford reference, for which a simplified diagram is presented below:



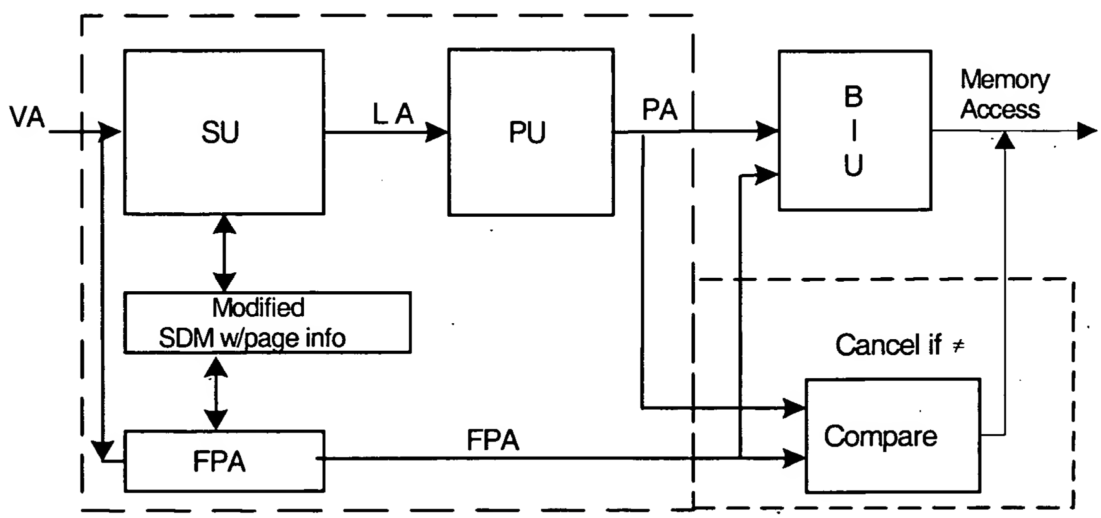

Crawford ATU

As seen above, each Virtual Address (VA) is converted by a segmentation unit (SU) during a first operation into a Linear Address (LA) using information from a segment descriptor memory (SDM). The LA is then in turn converted into a Physical address (PA) by a paging unit (PU) in a second operation and then used by a bus interface unit (BIU) to initiate a memory access. This set of operations takes place in the same slow, sequential manner for *every* virtual address, with no regard given to any characteristics of such address, prior address information, etc. It is always a two step operation, and, for each translation, takes the same exact time to complete; therefore it is inefficient in most cases.

The Applicant is the first to conceive a working solution that brings the speed benefits of the first type "integrated" segmentation/paging systems into the realm of the second type "independent" segmentation/paging systems where two distinct operations are

usually required to perform a memory access. It was not thought possible, until the applicant's invention, that a fast or speculative memory access could be performed in a separated segmentation/paging system without performing a full translation. To achieve this result, Applicant improved on the prior art by building in an additional parallel fast memory access path, where a tentative/speculative physical address is generated. As an example, the specification notes that this tentative/speculative physical address can be derived based on prior physical addresses that were used for performing prior memory accesses. The applicant's approach takes advantage of the fact that, between sequential memory accesses, it is unlikely that the physical page frame (which is on the order of 2K in size) will change. Thus, if this physical address information is saved, a later virtual address can be rapidly processed because the lower order portions of such address can be combined with such information to yield a fast or speculative physical address. To Applicant's knowledge, he was the first to formulate a workable mechanism/method that would allow tentative/speculative memory referencing in this type of a separated segmentation/paging system.

A simplified block diagram of the applicant's invention is presented below; please note that this is intended merely as an instructive tool to understand one embodiment of the claims, and should not be construed as limiting the same in any manner.




Applicant's ATU

Looking at the applicants' invention, in an Address Translation Unit (ATU) approach, after a first virtual address is converted, the physical address information from such conversion is kept in a modified segment descriptor memory, accessed by segment

identifier, which supplies this information to a fast physical address generator. Thereafter, each subsequent Virtual Address (VA) whose segment identifier is the same is converted by a fast physical address generator to create a tentative or speculative address used by a bus interface unit (BIU) to initiate a memory access. In other words, this memory access is initiated extremely rapidly, without waiting for the two-step translation noted above to complete. The *only* time the two-step calculated physical address is used to access memory is when a portion of such calculated physical address is different from the corresponding portion of the tentative or speculative address. When the two portions are different, the tentative memory access is discontinued by the comparator, and a memory access based on the fully calculated physical address is instead performed.

It is apparent that this tentative/speculative physical address is obtained much faster than the fully computed address generated by the conventional prior art approach. This is because there is no need to perform a 32 bit addition (to obtain the linear address from the virtual address) and no page number to page frame conversion (to obtain the physical address from the virtual address). In fact, as seen in FIG. 3A because only the lower order 12 bits of the linear address are used in the preferred embodiment, the fast address can be calculated before the full 32 bit linear address is even formed.

Thus, the present invention provides an accelerated memory reference that is compatible with the traditional two-step operation systems (i.e., such as those that are used in x86 instruction set systems using both segmentation and independent paging). This feature is extremely attractive since it means that such systems can also benefit from tentative/speculative memory referencing that was only previously available in so-called integrated segmentation/paging environments.

## 6. ISSUES

Are claims 38 – 112 obvious within the meaning of § 103 in light of Toy taken in combination with Crawford?

## 7. GROUPING OF CLAIMS

The claims have all been rejected under § 103 in light of Toy taken with Crawford. The claims do not, however, stand or fall together for the reasons because in general they are directed to different facets of the present inventions and/or are more particularly directed to

specific features of such inventions. A detailed discussion of such differences is given below.

8. ARGUMENT

As discussed in Section 8(A) below, Applicant explains in detail the teachings of the prior art, including the fact that the Toy reference does not teach or disclose a system that uses either: (1) a virtual address of the [segment identifier: segment offset] variety; (2) a linear address of the type used in the present invention; or (3) a virtual-linear address conversion operation. The Crawford reference, in turn, plainly does not show an accelerated or speculative memory reference.

Following this technical explanation in Section A, Applicant then demonstrates that the Examiner's application of Toy to Crawford is improper because: (1) there is nothing in either reference to suggest the combination; (2) there is no practical mechanism taught by Toy to a skilled artisan about how the fast address system could be employed within an environment like that shown in Crawford; and (3) the Crawford reference specifically teaches away from the combination suggested by the Examiner.

Finally, in section 8(B), Applicant reviews the claims individually and in-depth to explain how they are patentable over the aforementioned Toy and Crawford references.

A. Explanation of the Prior Art

1. The Toy reference is not an independent segmentation/paging system, and thus does not teach anything concerning an operation to convert a virtual address having a segment identifier/offset into a linear address, or a linear address into a physical address within a speculative addressing environment

Applicant acknowledges that the Toy reference discloses a form of fast memory reference, but it is set out in the context of a caching scheme, *not* an address translation system. This fast memory reference, furthermore does not take place in an environment like that set out in various forms in the present claims: i.e., with an independent segmentation/paging system, with linear addresses, with virtual-linear-physical operations, etc.

That this is true can be verified from the fact that the virtual address in Toy is specifically noted as having a format of the type [segment:page:word]. *See e.g.*, column 3, ll. 57+. This is clearly an integrated segment/paging format, and *not* a separated and independent segment/paging system.

From this discussion, too, it quite plain that Toy is not using the kind of virtual address of the [segment identifier: segment offset] type disclosed in Applicant's specification and recited in many of the claims, where a first part indicates a logical segment, and the second part an offset within such segment.

As there is already paging information in the Toy virtual address, this is *inconsistent* with an independent segmentation/paging system where virtual-linear and linear-physical operations occur. In fact, because segmentation and paging are integrated in Toy, the transformation from virtual to physical addresses is performed in one step. This is in turn confirmed by the explanation in Toy that the word address bits of this virtual address are not even translated. *See e.g.*, c.3 64 - 69. From this fact alone it can be seen that it is not a separated segmentation/paging system like in the present invention, *and thus there never is any "linear" address of any kind.*

The details of "how" Toy generates a fast address also seem particular to that system, and are not extendible in any apparent fashion to the kind of environment shown in Crawford. For one thing, Toy is not really *generating* a fast physical address; Toy simply reads the *untranslated lower* "word" bits of the *virtual address* (on line 115) and combines them with two *lower* order bits of a prior physical address (on line 119 from data latch 106 or instruction latch 105) as a lookup tag to a cache table to see if there is a match with a full physical address already stored there as an address tag word. *See e.g.*, column 4 – 5. The "lookup" tag itself is also neither a virtual, linear or physical address, nor a translated version of any of these. Seen from this perspective, therefore, it is difficult to even see how Toy is even "generating" a fast physical address, or a tentative/speculative address such as that disclosed in the present invention.

Nevertheless the more important reality is that the Toy approach of saving the lower two physical bits of the prior translated virtual address, and then combining it with some untranslated "word" portion of the virtual address to generate a speculative memory reference *would not work* within an independent segmentation/paging system. This is because, unlike Toy, there is no lower order "word" portion of a virtual address that goes

untranslated in an independent segmentation/paging system that can be used. The Toy approach is simply incompatible with this type of environment. And Toy notably does not include any further elaboration or explanation that this problem exists, and/or how it would be overcome.

To a large extent this is because, as alluded to earlier, Toy is really directed to a caching scheme; the actual address *translation* performed by ATB 102 is actually quite conventional. In fact, a careful review shows that there is no mention or suggestion in Toy that the translation from one virtual address to the next address is performed any differently, or any more quickly as in the present invention. All that happens in Toy is that some prior address bits are cached, but there is no "faster" second address translation. Thus, Toy does not include any suggestion that the translation process itself can be expedited.

These points are important because, as the Applicant's disclosure reveals, it is *not* a trivial matter to implement a tentative/speculative memory reference in an independent segmentation/paging system, or to implement a system where the actual translation process can be enhanced. Furthermore, the conventional wisdom in the art prior to the Applicant's invention was that it was impractical if not impossible to implement such functionality in the latter systems. Thus, it is apparent that the Examiner's evaluation of Toy is somewhat in error, and that such reference is far less instructive than he may have thought.

2. Toy does not teach or suggest anything to one of skill in the art about how Crawford could be modified to include accelerated memory reference capability.

Applicant acknowledges that the Crawford reference, unlike Toy, *is* an independent segmentation/paging based system. However, the Crawford reference makes no mention of a fast address generator, an accelerated or speculative memory referencing, etc., and there is no teaching or suggestion within Crawford to modify the same to include such functionality. In this regard, the Examiner is also apparently not contending that the Crawford reference itself contains any such teaching or suggestion.

Instead, the Examiner has indicated that it would be obvious to use Toy to modify the Crawford reference to include the type of capability discussed in former. The burden is on the PTO to produce sufficient evidence of prima facie obviousness. In re Rijckaert, 9 F.3d 1531, 1532, 28 USPQ 2d 1955, 1956 (Fed. Cir. 1993); In re Fine, 837 F.2d 1071, 1074, 5 USPQ 2d 1596, 1598 (Fed. Cir. 1988). It is apparent that a complete lack of evidence cannot

meet the criteria needed for a prima facie case of obviousness, and, as in the present case, the rejections should be overturned. *See e.g.*, Rijckaert 9 F. 3d at 1532; In re Neilson, 816 F.2d 1567, 1572, 2 USPQ 2d 1525, 1528 (Fed. Cir. 1984); In Re Gordon, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984).

In this regard, the present evidence is woefully inadequate to support an obviousness finding. The Examiner does not point out or explain where in Toy or the prior art that there is any hint or motivation that the Toy approach might be achievable or desirable in a Crawford type environment. As a preliminary matter, therefore, Applicant submits that the rejections to date under 103 are not supportable under applicable precedent, because the evidence of obviousness is profoundly deficient.

The other problem is that even if one adopts the Examiner's argument that it is obvious to use a Toy approach in Crawford, one of skill in the art would still be left to speculate and experiment about how to incorporate the Toy functionality in a completely different translation system environment, because neither reference even contemplates such an extension. As explained above, the Toy reference merely retains two physical address bits from a prior virtual address and combines them with an untranslated portion of the virtual address to perform a cache lookup; it is not instructing one skilled in the art about how to formulate a speculative memory reference for even an integrated segmentation/paging system, let alone an independent segmentation/paging system.

Thus, Applicant also cannot determine in what manner the Examiner is proposing that Crawford could be modified to work in the manner described in Toy. Furthermore, every conceivable scenario appears to lead to an unworkable mechanism.

For example, is the Examiner arguing that the segment descriptor memory in the segmentation unit of Crawford could store physical addresses and be subject to a lookup like the cache shown in Toy? This would be in express violation of the language in Crawford (see column 3, ll. 11 – 13) where it indicates that the segment tables *do not contain* any paging (physical address) information. Furthermore, as explained above, the virtual address in Crawford must be translated in its entirety, so the proposed approach by Toy could not be used as a look up in the Crawford environment.

If instead the Examiner is arguing that the page cache of Crawford could store physical addresses, then this would not result in a "fast" physical address since it cannot be

accessed until a linear address is first generated, and then mapped to a page number. Thus, this application of Toy to Crawford does not work either.

The Examiner is plainly relying on an "obvious to try" argument. This type of argument is usually not an acceptable basis for rejecting claims, especially where the technology is complex, the primary reference on its face is deficient, and the explanation provided is of a limited use in an environment the Applicant's. So, tere is simply no enlightening discussion presented in Toy that would lead one of skill in the art to a solution that would work in altogether different environment that requires completely different type of address processing.

3. The Crawford reference actually specifically teaches away from the combination now proposed by the Examiner

Finally, there is an even more compelling and objective reason why the combination proposed by the Examiner is not obvious, and that is the primary reference that the Examiner relies upon to be modified – Crawford – in fact specifically instructs against the type of approach disclosed in Toy. This is apparent from a number of observations.

First, the Crawford reference relies on a conventional segment descriptor. This entity is discussed at column 3, lines 11 –13:

"Segments are defined by a set of segment descriptor tables *that are separate from the page tables used to describe the page translation.*" (emphasis added)

Later on (column 10, ll. 9 – 14) it is specifically claimed that:

"...said segment descriptor table describing *segments without reference to whether a segment is paged or unpaged.*" (emphasis added)

The reasons for this is simple: Crawford is a separated segmentation/paging system, and this is highlighed by the demarcation in FIG. 2 with the dotted line separating the segmentation and paging units. There is no reasonable way to read Crawford that does not compel a conclusion that there is a very express prohibition that would dissuade those skilled in the art from using prior physical address information for a memory reference, and especially within a segmentation unit.

Thus, the Examiner's contention that it is obvious to try to modify Crawford to include the type of functionality of Toy is completely contradicted by the objective evidence of record. Moreover, when a reference teaches away from the kind of modification

proposed by the Examiner, a rejection for obviousness is not proper. See Fine, supra at 1599: "instead of suggesting that the system be used to detect nitrogen compounds, Eads deliberately seeks to avoid them; he warns against rather than teaches Fine's invention." Similarly, Gordon, supra at 112 reaches the same conclusion: "...if the French apparatus were turned upside down, it would be rendered inoperable for its intended purpose.....[I]n effect, French teaches away from the Board's proposed modification."

It is quite apparent that Crawford specifically and actively discourages one of skill in the art from modifying it in the fashion now proposed by the Examiner. The approach by Toy, is also quite incompatible with Crawford, and, this, reality too mitigates strongly against a suggestion that the combination is obvious.

Despite the fact that the prior art teaches away from such an approach, the Applicant went against the conventional wisdom and modified the descriptor disclosed in Crawford (see FIG. 2 of the present disclosure) to include some additional physical address information from a prior address translation. This is done in a way that allows fast tentative/speculative memory references while still preserving compatability with the conventional virtual-linear-physical addressing mechanisms disclosed in Crawfords. Thus, with a minimal amount of overhead the Applicant has provided a structure and method that significantly improves the memory access capability of independent segmentation/paging systems. The benefits of including some paging information during a segmentation process were completely unrealized until the Applicant's invention because the prior art expressly taught against such approach.

B. The Claims Are Patentable Over the Art of Record

1. The System of Claims 38 – 42 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 38 recites a system for performing address translations, including a combination of a means for generating an actual physical address, and a fast physical address generator. As further defined by the claim, the fast physical address generator generates a physical address faster (in less time) than the normal physical address generator.

In this claimed combination, the means for generating the physical address uses a virtual address "...having both a segment identifier and a segment offset." As noted above, Toy uses a type of virtual address that includes an integrated segment/paging structure; but it does not have a separate segment identifier and segment offset. Furthermore, in the combination of claim 38, a *linear address* is calculated based on the *entire virtual address.* As explained earlier, Toy does not calculate a linear address as that term is understood in the art, and defined in the specification. Moreover, even if the cache "lookup" tag that Toy uses could be considered a linear address (which it plainly is not) this tag is not based on the entire virtual address. Thus, there are significant differences articulated in the claim to distinguish over Toy.

Finally, there is no reasonable hint or suggestion in Toy that it could be used to modify an address translation system such as that shown in Crawford to arrive at the claimed combination. Even if it were otherwise desirable to do so, the Toy teachings are far too deficient to enlighten one skilled in the art about how to go about doing so. And, given the strong suggestion in Crawford that it should not include physical address information at the early stage of address translation, one skilled in the art would be surely led away from making the kind of combination now described in claim 38. The technical reasons for this were explained in detail above. To date, the Examiner has failed to address these deficiencies in Toy, and has failed to explain why one skilled in the art would disregard the express teachings of Crawford to modify it in a manner that is incompatible with such system.

Accordingly, Applicant submits that claim 38, and its dependent claims 39 – 42, are clearly patentable over the prior art of record.

2. The System of Claims 43 - 48 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Like claim 38, claim 43 also recites a system for performing address translations that makes use of a tentative physical address, and also uses a virtual address having both a segment identifier portion and a segment offset portion, in which a linear address is computed during a first operation based on all portions of the virtual address. Accordingly, it distinguishes over the art of record for at least the same reasons as claim 38.

Claim 43 further recites, however, that the tentative physical address can be generated "....before a second operation has completed converting said first linear address into a physical address." There is no mention or suggestion anywhere in Toy of how a system like Crawford would be modified, let alone that it would behave in the manner described in this limitation of claim 43. This is because Toy does not use a sequential set of virtual – linear and linear – physical operations as are required in independent paging systems, and says nothing at all about how to handle the timing or sequencing of the same vis-à-vis a tentative memory reference. The Examiner has not explained how this glaring gap in the Toy teachings can be reconciled by a skilled artisan in any reasonable fashion. Consequently, the invention of claim 43 cannot be obvious to one skilled in the art.

Dependent claim 46 further distinguishes on the basis that it specifically details that the tentative physical address is generated "based on a combination of prior physical address information and partial linear address information relating to said first virtual address." There is nothing in Toy that suggests that Crawford would be modified in this way, or that this particular mechanism for generating the tentative address would have been obvious to one skilled in the art.

Furthermore, dependent claim 47 further limits the timing aspect of the tentative physical address generation, by reciting that it occurs "... before said first operation has completed converting said first virtual address into said first linear address. " Again, this particular type of sequencing and timing is not discussed at all in Toy, as it is an integrated (one step) and not a two step sequential (virtual-linear, linear-physical) system. Thus, this approach is beyond any explicit and/or implied teachings of Toy.

Finally, claim 48 notes that the "...virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address." As Toy has nothing to say about linear addresses, it cannot instruct or suggest to anyone skilled in the art to modify Crawford to include the kind of nuance disclosed in this claim.

3. The System of Claims 49 - 53 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 49 is directed to a computer system which performs address translations using a first operation to convert virtual addresses having both a segment identifier portion and a segment offset portion to linear addresses, such that both the segment identifier and segment offset portions of the virtual addresses are used for converting said linear addresses and a second operation to convert said linear addresses to physical addresses. This computer system also includes a fast physical address generator for generating fast physical addresses related to the virtual addresses.

The discussion above for claim 38 applies with equal force to this portion of the claim, and so it will not be repeated here.

But claim 49 further distinguishes on the basis that it recites that the fast physical addresses can be generated "....while or before said virtual addresses are converted in said first operation into said linear addresses." This aspect of claim 49 has never been addressed by the Examiner, and points out another aspect of the invention that distinguishes fairly clearly over any teaching or creative elaboration attributable to Toy. First, the Examiner has apparently argued that the fast physical addresses are "generated" in Toy as a result of locating entries with a tag lookup in the tag memory 108. But it is equally apparent this can only happen after (not before) the tag lookup is created in the first place. Thus, even if the Examiner equates this tag with a linear address, one following the Toy teachings would not modify Crawford to behave in the manner called for in the claim. Again, the Examiner has not addressed this contradiction in his own logic.

This operational discrepancy points out yet another significant deficiency in the Toy reference, and why it would not be terribly instructive to one skilled in the art trying to modify a completely different translation environment (independent segmentation and paging). One skilled in the art would, in all likelihood, be led away by Toy from conceiving of the type of system articulated in claim 49, and therefore this claim cannot be obvious in light of such reference.

Dependent claims 52 and 53 recite details of the invention that parallel those set forth in claims 46 – 48, and thus distinguish for those reasons as well.

4. <u>The System of Claims 54 - 56 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford</u>

Independent Claim 54 is directed to a system for performing address translations, and it includes a virtual to linear address converter circuit for generating a calculated linear address based on a virtual address, said virtual address having both a segment identifier and a segment offset, and said calculated linear address being based on all of said virtual address. This is in combination with a linear to physical address converter circuit for generating a calculated physical address based on the calculated linear address, the calculated physical address including a calculated page frame and a calculated page offset, and a fast physical address generator.

The discussion above for claim 38 applies with equal force to this portion of the claim, and so it will not be repeated here.

Claim 54 further distinguishes on the basis that the fast physical address includes "..a fast page frame and a fast page offset." There is simply nothing in <u>Toy</u> that teaches or suggests to one skilled in the art that he/she should modify <u>Crawford</u> to include a fast physical address having this form (or any other form for that matter).

Dependent claim 56 further specifies that the virtual address is "...partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address." Again, <u>Toy</u>, shows nothing about linear addresses, let alone anything about "partially" converting a linear address in the manner called for this in this claim to generate a tentative physical address.

5. <u>The System of Claims 57 - 60 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford</u>

Claim 57 is also directed to a system for performing address translations. This is done using a first operation to convert a first virtual address having both a segment identifier portion and a segment offset portion to a first linear address, such that all portions of the virtual address are considered when converting said virtual address into the first linear address and a second operation to convert said first linear address to a first physical address. These features distinguish for the same reasons as articulated for claim 38 above.

Moreover, claim 57 specifies that the system further includes an address translation memory, accessible by said system while said first operation is converting said first virtual address, and capable of storing prior physical address information generated during a prior address translation by said second operation based on a prior virtual address. As noted earlier, the Crawford reference specifically indicates that the segment descriptor memory in the segmentation unit (which performs virtual to linear translations) should not contain physical address information. Thus, even if Crawford were modified by Toy, it cannot be modified to include a kind of "address translation memory" that is used during the first operation (virtual to linear) to store "prior physical address information" as called for in this claim.

Finally, claim 57 further indicates that the "...fast physical address can be generated based on the prior physical address information and said first linear address before said second operation has completed converting said first linear address." Again, as articulated above for claim 49, Toy says and suggests nothing to one skilled in the art about how the timing for a independent segmentation and paging system such as Crawford should be configured for a fast physical address where there are also virtual-linear and linear-physical operations.

For similar reasons to those articulated above for claim 54, dependent claim 60 - which indicates that the fast physical address is comprised of a page frame portion based on the prior physical address information and a page offset portion based on the result of converting said first virtual address to a first linear address – also is patentable over the art of record.

6. The System of Claims 61 - 65 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 61 is patentable for the same reasons as claim 38, and for several other reasons. For example, this claim mentions a "segment base" associated with the virtual address (something that Toy does not show) and the fact that the fast physical address is "...comprised of the stored page frame combined with a fast page offset portion derived from the segment base and the virtual address." This claim points out yet another example of where the Examiner has improperly rejected a claim on the bald contention that Toy "suggests" this kind of specific modification to Crawford, without ever pointing out where

this teaching is supposedly found. The reason for this clear to see: there is none, and the Examiner's rejection is unfounded because it relies much too heavily on speculation about one skilled in the art would do to Crawford after considering the Toy reference. This is not a proper basis for an obviousness rejection.

Like claim 47 above, dependent claim 64 further distinguishes because there is simply no suggestion or teaching in Toy about how to configure the relative timing of fast address and regular address operations.

7. The Method of Claims 66 - 69 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 66 is directed to a method of performing translation of a virtual address in a computer system using segmentation and optional independent paging. To accomplish this, the method recites that the following steps are performed:

(a) calculating a fast physical address related to said virtual address; and

(b) calculating a linear address based on said virtual address, said linear address being based on both a segment identifier and segment offset portion of said virtual address; and

(c) calculating an actual physical address based on the linear address;

wherein step (a) is completed prior to the completion of step (c), and the fast physical address can be used to initiate a fast memory reference.

In a similar fashion as explained earlier for claim 38, the virtual address of Toy does not have a segment identifier and a segment offset, so Toy cannot perform the step noted in (b). Nor does Toy contain any suggestion to one skilled in the art to perform step (a) (fast physical address) in an environment in which steps (b) and (c) take place, or how this would occur. Accordingly, this claim is not made obvious by the art of record.

Again, for the reasons set out in claim 64, dependent claim 69 further distinguishes in that it recites that "...step (a) is completed prior to the completion of step (b)."

8. The Method of Claims 70- 73 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 70 is directed to a method of generating memory references based on virtual addresses in a computer system, where the computer system is using segmentation and optional independent paging. The method includes the steps of:

(a) generating tentative memory references based on said virtual addresses; and

(b) converting said virtual addresses to linear addresses during a segmentation operation, said linear addresses being based on translating all portions of said virtual address; and

(c) converting said linear addresses to physical addresses during a paging operation, so that actual memory references can be made based on said physical addresses;

wherein the tentative memory reference can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

It should be apparent at this point that this claim distinguishes for the same reasons as set out above for claims 38, etc. Furthermore, as explained for claim 49, the Examiner's rejection here is *inconsistent* with his explanation of Toy, since the fast memory reference there is performed after (not before) the tag lookup is created in the first place. Thus, even if the Examiner equates this tag with a linear address, one following the Toy teachings could not modify Crawford to behave in the manner called for in the claim, which says that the tentative memory reference "...can be generated *while* said virtual addresses are being converted in said first operation into said linear addresses."

Furthermore, for the reasons set out in claim 64, 69, dependent claim 73 further distinguishes in that it recites that "...step (a) is completed prior to the completion of step (b)."

9. The Method of Claims 74 - 76 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 74 is directed to a method of generating a fast memory reference using a fast physical address derived from a virtual address having both a segment identifier and a segment offset in a computer system employing both segmentation and optional independent paging. The method includes the steps of:

(a) converting a portion of said virtual address into a partial linear address; and

(b) combining the partial linear address with physical address information obtained from a prior memory reference to generate said fast physical address;

(c) generating a memory reference based on the fast physical address;

(d) converting said virtual address into an actual physical address during which time a linear address is also calculated based on both the segment id and segment offset of said virtual address;

(e) cancelling the memory reference if the fast physical address and actual physical address are different.

This claim is patentable over the art, for the same reasons as set forth above for claims 38 as for the limitations in the claim pertaining to "segment identifier," "segment offset," "linear address," etc., and for the fact that there is no mention anywhere in Toy that a "partial linear address" should exist (per step (a)) let alone be combined with physical address information as set out in step (b).

For dependent claims 75, 76, as explained above also for claim 73 (and others) there is also no teaching or suggestion that the fast physical address is generated prior to the generation of the linear address; in fact, this is incompatible with the teachings of Toy.

10. The Method of Claims 77- 81 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 77 covers a method of generating physical addresses from virtual addresses in a computer system employing both segmentation and optional independent paging. The method includes the steps of:

(a) generating a first calculated linear address based on a first virtual address in a first operation, said linear addresses being based on translating all portions of said first virtual address; and

(b) generating a fast physical address in a second operation, the fast physical address including linear address information relating to said first virtual address and portions of physical address information relating to said first virtual address; and

(c) generating a first calculated physical address in a third operation based on the first calculated linear address;

wherein the fast physical address is generated prior to the generation of the first calculated physical address.

As with claim 74, this claim is patentable over the art, for the same reasons as set forth above for claims 38 as for the limitations in the claim pertaining to "segment identifier," "segment offset," "linear address," the fact that all portions of the linear address are translated (step (a)), etc.

Dependent claim 81 further differentiates by virtue of the fact that it covers those embodiments where "...the first and second operations overlap in time, and the fast physical address is generated prior to the generation of the first calculated linear address." Again, as explained for claim 76 above (and others), there is simply nothing in Toy that would lead one of skill in the art to use this type of approach.

11. The System of Claims 82 - 85 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 82 covers a system for performing memory references in a processor which employs both segmentation and optional independent paging during an address translation. Again, like claim 38, it requires that there be a virtual address having a "segment identifier" and "offset" and that all of the virtual address is translated. For this reason, the claim is patentable as written.

Furthermore, as explained earlier, Toy only describes an embodiment where a portion of the virtual address is left untranslated, and this portion is used to find a fast physical address. He simply gives no guidance or suggestion to one skilled in the art about how to modify a system like Crawford's, so that a fast memory reference could occur as it does in claim 82 – i.e., based on translating the *entire* virtual address.

Accordingly, the rejection of this claim, and its dependent claims (83 – 85) is also not proper at this time.

12. The Method of Claims 86 - 88 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 86 covers a method for performing memory accesses between a processor and a memory, where the processor has an address translation mechanism that employs

segmentation and optional independent paging. For the most part, the language of this claim tracks that of claim 82, and accordingly the same reasoning would apply with equal force here to this claim and its dependent claims.

13. <u>The System of Claims 89 - 94 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford</u>

Claim 89 is also directed to a system for performing address translations, and is distinguishable over the art for many of the same reasons as set forth above.

In addition, however, this claim specifically recites that the "...second address translation can be performed faster than the first address translation," and this further differentiates in an non-obvious manner over the art. A careful review shows that there is no mention or suggestion in <u>Toy</u> that the translation from one virtual address to the next address is performed any more quickly. All that happens in <u>Toy</u> is that some prior address bits are cached, but there is no "faster" second address translation. Thus, <u>Toy</u> does not include any suggestion that the translation process itself can be expedited. This can be confirmed with reference to the description of the ATB 102 in <u>Toy</u>, which, as described, *always performs the same operation* (i.e., a single transformation from virtual to physical) from one virtual address to the next. There is simply no mechanism provided for "translating" one virtual address any faster than the next, and this critical distinction for this claim has also been ignored by the Examiner.

As noted in claim 94, a register is also used "...for storing address information pertaining to the first virtual address for use during said translation of said second virtual address." Nowhere in <u>Toy</u> is there any indication that virtual address information is retained for purposes of performing another *translation* as set out in this claim.

14. <u>The System of Claims 95 - 100 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford</u>

In similar fashion to claim 89, claim 95 claims a circuit for performing fast translations of virtual addresses to physical addresses in a computer system which uses both segmentation and optional independent paging. This claim differentiates over the art for

many of the same reasons already expressed for claims 38 – 94. Moreover, as with claim 89, it is further explained that the address generator uses information from the first address translation during the fast address translation so that the *translation* of the second virtual address takes less time than the first address translation. As demonstrated above, Toy does not perform a second "translation" any faster than the first translation. The invention of claim 95 accomplishes this result because, as set out in the claim, unlike for the first virtual address, the second virtual address is *not* completely converted into a linear address for the fast address. There is simply nothing like this described in Toy or Crawford.

For dependent claim 100, which recites that there is a "...register for storing address information pertaining to the first virtual address for use during said translation of said second virtual address" the same analysis as for claim 94 would apply, compelling the conclusion that this claim is also non-obvious and in allowable form.

15. The Method of Claims 101 - 106 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 101 also covers a method of translating virtual addresses in a computer system that uses both segmentation and optional independent paging. For the most part, the same discussion above pertaining to claims 38 and 49 applies with equal force here. Thus, the Applicant submits that this claim, and its dependent claims (102 – 106) are clearly patentable over the art.

16. The Method of Claims 107 - 112 Is Not Disclosed or Made Obvious by the Combination of Toy and Crawford

Claim 107 also covers a method of performing address translations in a computer systemthat uses both segmentation and optional independent paging. The same reasoning as for claims 38, 49, etc. apply here, as well as the discussion for claim 89 above, since this claim specifically indicates that the "second address translation" is achieved in less time than the "first address translation." Applicant submits that one skilled in the art would not know how to modify Crawford to achieve this functionality, since there is not even any mention of how to accomplish this kind of result even in the Toy environment.

For dependent claim 112, the same discussion as for claim 106 (and others) would apply here as well.

10. <u>CONCLUSION</u>

Applicant submits that the Examiner has seriously misunderstood the teachings and limitations of the <u>Toy</u> reference. Many aspects of the claims have been overlooked by the Examiner, and he has not provided any evidence that such limitations are actually taught, suggested, or even hinted at by the reference. Such phantom teachings cannot be "combined" in any fashion with <u>Crawford</u> to arrive at the claimed combinations. Thus, the present rejections are not sustainable with the evidence of record.

Furthermore, despite the fact that the prior art teaches away from the present claims, the Examiner insists on applying aspects of the <u>Toy</u> reference that are completely inconsistent and inapplicable to the kind of system shown in <u>Crawford</u>. The objective indicia evidence, therefore overwhelmingly points to the fact that the claims would not have been obvious to a person skilled in the art.

On these bases, the Applicant respectfully requests that the present rejections of the claims be withdrawn, and that the claims be allowed in their present form.

Date: March 20, 2000

J. Nicholas Gross

J. Nicholas Gross, Attorney, Reg. No. 34,175

# APPENDIX 1 - PENDING CLAIMS

38. A system for performing address translations usable by a processor employing both segmentation and optional independent paging the system comprising:

means for generating an actual physical address from a virtual address in a time period T, said virtual address having both a segment identifier and a segment offset by calculating a linear address based on said entire virtual address, and by calculating said actual physical address based on said calculated linear address; and

a fast physical address generator for generating a fast physical address related to said virtual address in a time < T.

39. The system of claim 38, wherein the fast physical address can be used for generating a memory access faster than a memory access based on said actual physical address.

40. The system of claim 39, including a cancellation circuit for cancelling the memory access if the fast physical address and actual physical address are different.

41. The system of claim 38, wherein the fast physical address is generated based on a combination of physical address information from a different virtual address, and partial linear address information relating to said virtual address.

42. The circuit of claim 38, wherein the fast physical address is generated before said calculated linear address.

# APPENDIX 1 - PENDING CLAIMS

43. A system for performing address translations using a first operation to convert a first virtual address having both a segment identifier portion and a segment offset portion to a first linear address, the first linear address being based on all portions of the virtual address and a second operation to convert said first linear address to a first physical address, said system further including:

a tentative physical address generator for generating a tentative physical address related to said first virtual address;

wherein the tentative physical address can be generated before said second operation has completed converting said first linear address.

44. The system of claim 43, wherein the tentative physical address can be used for generating a memory access which is faster than a memory access resulting from said first physical address.

45. The system of claim 44, including a cancellation circuit for cancelling the memory access if the tentative physical address and first physical address are different.

46. The system of claim 43, wherein the tentative physical address is generated based on a combination of prior physical address information and partial linear address information relating to said first virtual address.

47. The circuit of claim 43, wherein the tentative physical address is generated before said first operation has completed converting said first virtual address into said first linear address.

48. The circuit of claim 43, wherein said first virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address.

# APPENDIX 1 - PENDING CLAIMS

49. A computer system which performs address translations using a first operation to convert virtual addresses having both a segment identifier portion and a segment offset portion to linear addresses, such that both the segment identifier and segment offset portions of the virtual addresses are used for converting said linear addresses and a second operation to convert said linear addresses to physical addresses, said system further including:

a fast physical address generator for generating fast physical addresses related to said virtual addresses;

wherein the fast physical addresses can be generated while or before said virtual addresses are converted in said first operation into said linear addresses.

50. The system of claim 49, wherein the fast physical addresses can be used for generating memory accesses faster than memory accesses resulting from said calculated physical addresses.

51. The system of claim 50, including a cancellation circuit for cancelling the memory accesses if the fast physical addresses and calculated physical addresses are different.

52. The system of claim 49, wherein the fast physical addresses are generated based on a combination of physical address information and partial linear address information relating to said virtual addresses.

53. The system of claim 49, wherein said virtual addresses are partially converted to linear addresses by the fast physical address circuit and are combined with physical address information relating to prior virtual addresses to generate the tentative physical addresses.

# APPENDIX 1 - PENDING CLAIMS

54. A system for performing address translations comprising:

a virtual to linear address converter circuit for generating a calculated linear address based on a virtual address, said virtual address having both a segment identifier and a segment offset, and said calculated linear address being based on all of said virtual address; and

a linear to physical address converter circuit for generating a calculated physical address based on the calculated linear address, the calculated physical address including a calculated page frame and a calculated page offset; and

a fast physical address circuit for generating a fast physical address including a fast page frame and a fast page offset;

wherein a memory reference can be generated based on the fast physical address.

55. The system of claim 54, wherein the fast physical address is based on linear address information relating to the virtual address and physical address information relating to a prior virtual address.

56. The system of claim 54, wherein the virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address.

# APPENDIX 1 - PENDING CLAIMS

57. A system for performing address translations using a first operation to convert a first virtual address having both a segment identifier portion and a segment offset portion to a first linear address, such that all portions of the virtual address are considered when converting said virtual address into the first linear address and a second operation to convert said first linear address to a first physical address, the system further including:

an address translation memory, accessible by said system while said first operation is converting said first virtual address, and capable of storing prior physical address information generated during a prior address translation by said second operation based on a prior virtual address;

wherein a fast physical address can be generated based on the prior physical address information and said first linear address before said second operation has completed converting said first linear address.

58. The system of claim 57, wherein the fast physical address can be used for an accelerated memory access which is faster than a memory access resulting from said first physical address.

59. The system of claim 58, including a cancellation circuit for cancelling the fast memory access if the fast physical address and first physical address are different.

60. The system of claim 57, wherein the fast physical address is comprised of:

(iii) a page frame portion based on the prior physical address information; and

(iv) a page offset portion based on the result of converting said first virtual address to a first linear address.

# APPENDIX 1 - PENDING CLAIMS

61. A computer system using segmentation and optional independent paging for performing address translations comprising:

an address translation memory capable of storing:

(i) a portion of a physical address corresponding to a stored page frame; and

(ii) segment base information relating to a virtual address; and

a virtual to linear address converter circuit for generating a calculated linear address based on combining a segment offset portion of the virtual address and the segment base, wherein all of said virtual address is used for generating the calculated linear address; and

a linear to physical address converter circuit for receiving and generating a calculated physical address based on the calculated linear address, the calculated physical address including a first page frame and a first page offset; and

a fast physical address circuit for generating a fast physical address comprised of the stored page frame combined with a fast page offset portion derived from the segment base and the virtual address;

wherein the fast physical address is calculated prior to the generation of said calculated physical address.

62. The system of claim 61, wherein the fast physical address can be used for generating a fast memory access which is generated more quickly than a memory access resulting from said first physical address.

63. The system of claim 61, including a cancellation circuit for cancelling the fast memory access if the fast physical address and first physical address are different.

64. The circuit of claim 61, wherein the fast physical address is generated prior to the generation of the first linear address.

65. The system of claim 61, wherein the stored page frame is generated in a prior address translation based on a prior virtual address.

# APPENDIX 1 - PENDING CLAIMS

66. A method of performing a translation of a virtual address in a computer system using segmentation and optional independent paging, said method including the steps of :

(a) calculating a fast physical address related to said virtual address; and

(b) calculating a linear address based on said virtual address, said linear address being based on both a segment identifier and segment offset portion of said virtual address; and

(c) calculating an actual physical address based on the linear address;

wherein step (a) is completed prior to the completion of step (c), and the fast physical address can be used to initiate a fast memory reference.

67. The method of claim 66, further including a step (d): cancelling the memory access if the fast physical address and actual physical address are different.

68. The method of claim 66, wherein the fast physical address is generated based on a combination of physical address information from a different virtual address, and partial linear address information relating to said virtual address.

69. The method of claim 66, wherein step (a) is completed prior to the completion of step (b).

# APPENDIX 1 - PENDING CLAIMS

70. A method of generating memory references based on virtual addresses in a computer system, said computer system using segmentation and optional independent paging, the method including the steps of:

(a) generating tentative memory references based on said virtual addresses; and

(b) converting said virtual addresses to linear addresses during a segmentation operation, said linear addresses being based on translating all portions of said virtual address; and

(c) converting said linear addresses to physical addresses during a paging operation, so that actual memory references can be made based on said physical addresses;

wherein the tentative memory reference can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

71. The method of claim 70, further including a step (d): cancelling the tentative memory reference if the tentative memory reference and actual memory reference are different.

72. The method of claim 70, wherein the tentative memory reference is generated based on a combination of physical address information and partial linear address information relating to said virtual addresses.

73. The method of claim 70, wherein step (a) is completed prior to the completion of step (b).

# APPENDIX 1 - PENDING CLAIMS

74. A method of generating a fast memory reference using a fast physical address derived from a virtual address having both a segment identifier and a segment offset in a computer system employing both segmentation and optional independent paging, the method including the steps of:

(a) converting a portion of said virtual address into a partial linear address; and

(b) combining the partial linear address with physical address information obtained from a prior memory reference to generate said fast physical address;

(c) generating a memory reference based on the fast physical address;

(d) converting said virtual address into an actual physical address during which time a linear address is also calculated based on both the segment id and segment offset of said virtual address;

(e) cancelling the memory reference if the fast physical address and actual physical address are different.

75. The method of claim 74, wherein the fast physical address is generated prior to the generation of the linear address.

76. The method of claim 74, wherein the fast physical address is used to generate a fast memory access prior to the generation of the linear address.

# APPENDIX 1 - PENDING CLAIMS

77. A method of generating physical addresses from virtual addresses in a computer system employing both segmentation and optional independent paging, the method including the steps of:

(a) generating a first calculated linear address based on a first virtual address in a first operation, said linear addresses being based on translating all portions of said first virtual address; and

(b) generating a fast physical address in a second operation, the fast physical address including linear address information relating to said first virtual address and portions of physical address information relating to said first virtual address; and

(c) generating a first calculated physical address in a third operation based on the first calculated linear address;

wherein the fast physical address is generated prior to the generation of the first calculated physical address.

78. The method of claim 77, wherein the fast physical address is used to generate a tentative memory access prior to the generation of the first calculated physical address.

79. The method of claim 78, including a step (d): cancelling the tentative memory access if the fast physical address and first calculated physical address are different.

80. The method of claim 79, further including a step (e): generating a memory access request based on the first calculated physical address; and (f) storing physical address information relating to the first calculated physical address for use in a later address translation.

81. The method of claim 77, wherein the first and second operations overlap in time, and the fast physical address is generated prior to the generation of the first calculated linear address.

# APPENDIX 1 - PENDING CLAIMS

82. A system for performing memory references in a processor which employs both segmentation and optional independent paging during an address translation, said system comprising:

means for performing an address translation by generating a first physical address from a first virtual address by first calculating a first linear address based on both a first segment identifier and first offset associated with the first virtual address, such that all of said first virtual address is translated, and then calculating the first physical address based on the first calculated linear address; and

a fast physical memory access circuit for generating a fast memory reference, which fast memory reference is based on physical address information from said address translation means;

a bus interface circuit for initiating a fast memory access to a memory subsystem based on said fast memory reference.

83. The system of claim 82, further including a comparator for determining whether said fast memory reference can be used for a fast memory access.
84. The system of claim 83, further including a cancellation circuit for canceling said fast memory access.
85. The system of claim 84, wherein the system performs an actual memory reference after said fast memory reference is cancelled.

# APPENDIX 1 - PENDING CLAIMS

86. A method for performing memory accesses between a processor and a memory, said processor having an address translation mechanism that employs segmentation and optional independent paging, the method comprising the steps of:

generating computed physical addresses by converting virtual addresses having a segment identifier and a segment offset into linear addresses, such that all portions of said virtual addresses are translated, and then converting said linear addresses into a physical addresses;

generating a speculative physical address based on one of said computed physical addresses;

initiating a speculative memory access based on said speculative physical address.

87. The method of claim 86, further including a step of initiating an actual memory access based on a physical address which has been computed during separate segmentation and paging operations.
88. The method of claim 87, wherein said speculative memory access is completed unless canceled in favor of an actual memory access.

# APPENDIX 1 - PENDING CLAIMS

89. A system for performing a first and a second address translation of first and second virtual addresses respectively, the system comprising:

a virtual to linear address converter circuit for generating a first calculated linear address based on translating all portions of the first virtual address including a segment identifier and a segment offset~~,~~ and

a linear to physical address converter circuit for completing the first address translation by generating a first calculated physical address based on said first calculated linear address, said first calculated physical address including a first calculated page frame and a first calculated page offset; and

wherein the system uses information from the first address translation during the second address translation so that the second address translation can be performed faster than the first address translation.

90. The system of claim 89, further including a comparator for determining whether said second address translation can be used for a memory access.
91. The system of claim 89, wherein said second address translation is based on a combination of partial linear address information relating to said second virtual address and physical address information from a different virtual address.
92. The system of claim 89, wherein the system also calculates an actual second physical address from said second virtual address, by calculating a second linear address based on a second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.
93. The system of claim 92, wherein at least a portion of said actual second physical address is compared with a corresponding portion of said second physical address from said fast physical address generator, and when said portions are not equal, said actual second physical address is used for a memory access.
94. The system of claim 89, further including a register for storing address information pertaining to the first virtual address for use during said translation of said second virtual address.

# APPENDIX 1 - PENDING CLAIMS

95. A circuit for performing fast translations of virtual addresses to physical addresses in a computer system which uses both segmentation and optional independent paging, the circuit including:

an address generator for performing a first address translation of a first virtual address having an associated first segment identifier and a first offset, said first translation including converting all of said virtual address into a first linear address;

said address generator also performing a fast address translation of a second virtual address having an associated second segment identifier and a second offset, said fast address translation occurring without converting all of said second virtual address into a second linear address;

wherein said address generator uses information from the first address translation during the fast address translation so that said translation of said second virtual address takes less time than said first address translation.

96. The system of claim 95, further including a comparator for determining whether said fast address translation can be used for a memory access.
97. The system of claim 95, wherein said fast address translation is achieved based on a combination of partial linear address information relating to said second virtual address and physical address information from said first virtual address.
98. The system of claim 95, wherein the address generator also performs a calculated translation to calculate an actual second physical address from said second virtual address, by calculating a second linear address based on said second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.
99. The system of claim 98, wherein at least a portion of said actual second physical address is compared with a corresponding portion of said second physical address from said fast physical address generator, and when such portions are not equal, said actual second physical address is used for a memory access.
100. The system of claim 95, further including a register for storing address information pertaining to the first virtual address for use during said translation of said second virtual address.

# APPENDIX 1 - PENDING CLAIMS

101. A method of translating virtual addresses in a computer system that uses both segmentation and optional independent paging, the method including the steps of:

(a) generating a first calculated physical address based on a first virtual address in a first operation, said first virtual address including a first segment identifier and a first offset and wherein said first operation converts all of said virtual address into a first linear address; and

(b) generating a second fast physical address in a second operation based on a second virtual address, said second virtual address including a second segment identifier and a second offset, and said second fast physical address being generated based on information obtained during said first operation, and without converting all of said second virtual address into a second linear address;

wherein said second operation is performed faster than said first operation.

102. The method of claim 101, further including a step of determining whether a memory access can be made using said second fast physical address.

103. The method of claim 101, wherein during step (b) said second physical address is generated based on a combination of partial linear address information relating to said second virtual address and physical address information from said first virtual address.

104. The method of claim 101, further including a step (c): generating an actual second physical address from said second virtual address during a third operation, by calculating a second linear address based on said second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.

105. The method system of claim 104, further including step (d): comparing at least a portion of said actual second physical address with a corresponding portion of said second physical address from said fast physical address generator, and when such portions are not equal, using said actual second physical address for a memory access.

106. The system of claim 101, further including a step of storing address information pertaining to the first virtual address in a register during said first operation for use during said second operation.

# APPENDIX 1 - PENDING CLAIMS

107. A method of performing address translations in a computer systemthat uses both segmentation and optional independent paging, the method including the steps of:

(a) performing a first address translation by translating a first virtual address into a first physical address by: (i) first calculating a first linear address based on a first segment identifier and first offset associated with said first virtual address wherein all of said virtual address is translated; and (ii) calculating said first physical address based on said first calculated linear address and

(b) performing a second address translation using information obtained during said first address translation to translate a second virtual address into a second physical address, said second physical address being obtained without converting all of said second virtual address into a second linear address;

wherein said second translation can be achieved in less time than said first translation.

108. The method of claim 107, further including a step of determining whether a memory access can be made using said second physical address.

109. The method of claim 107, wherein during step (b) said second physical address is generated based on a combination of partial linear address information relating to said second virtual address and physical address information from said first virtual address.

110. The method of claim 107, further including a step (c): generating an actual second physical address from said second virtual address, by calculating a second linear address based on said second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.

111. The method system of claim 110, further including step (d): comparing at least a portion of said actual second physical address with a corresponding portion of said second physical address from said fast physical address generator, and when such portions  are not equal, using said actual second physical address for a memory access.

112. The system of claim 107, further including a step of storing address information pertaining to the first virtual address in a register for use during said second address translation.

ORIGINAL JDG
RAB97-002

*Express Mail Label No. EG214043430US*

# Improved Address Translation Mechanism and Method in a Computer System

## FIELD OF THE INVENTION

The invention relates to the field of address translation for memory management in a computer system.

## BACKGROUND OF THE INVENTION

Advanced computer hardware systems operate with complex computer software programs. Computer system designers typically separate the virtual address space, the address space used by programmers in their development of software, and the physical address space, the address space used by the computer system. This separation allows programmers to think in terms of their conceptual models, and to design computer software programs without reference to specific hardware implementations. During the actual execution of programs by the computer system, however, these separate addresses must be reconciled by translating software program virtual addresses into actual physical addresses that can be accessed in a computer memory subsystem.

There are many well known approaches for address translation in the memory management mechanism of a computer system. These approaches fall into basically two major categories: those which map the smaller virtual (sometimes called logical, symbolic or user) addresses onto larger physical or real memory addresses, and those which map larger virtual addresses onto smaller physical memory. Translation mechanisms of the former category are employed typically in minicomputers in which relatively small address fields (e.g.: 16 bit addresses) are mapped onto larger real memory. Translation mechanisms of the second category are used typically in microprocessors, workstations and mainframes. Within each of these categories, segmentation only, paging only, and a combination of segmentation and paging are well known for accomplishing the translation process.

The present invention is primarily directed to address translation mechanisms where larger virtual addresses are mapped onto smaller physical addresses, and further to systems where segmentation and optional paging is employed.

In a segmentation portion of an address translation system, the address space of a user program (or programs cooperatively operating as processes or tasks), is regarded as a collection of segments which have common high-level properties, such as code, data, stack, etc. The segmented address space is referenced by a 2-tuple, known as a virtual address, consisting of the following fields: <<s>,<d>>, where <s> refers to a segment number (also called identifier or locator), and <d> refers to a displacement or offset, such as a byte displacement or offset, within the segment identified by the segment number. The virtual address <17,421>, for example, refers to the 421st byte in segment 17. The segmentation portion of the address translation mechanism, using information created by the operating system of the computer system, translates the virtual address into a linear address in a linear address space.

In a paging portion of an address translation system, a linear (or intermediate) address space consists of a group of pages. Each page is the same size (*i.e.* it contains the same number of addresses in the linear space). The linear address space is mapped onto a multiple of these pages, commonly, by considering the linear address space as the 2-tuple consisting of the following fields: <<page number >,< page offset>>. The page number (or page frame number) determines which linear page is referenced. The page offset is the offset or displacement, typically a byte offset, within the selected page.

In a paged system, the real (physical) memory of a computer is conceptually divided into a number of page frames, each page frame capable of holding a single page. Individual pages in the real memory are then located by the address translation mechanism by using one or more page tables created for, and maintained by, the operating system. These page tables are a mapping from a page number to a page frame. A specific page may or may not be present in the real memory at any point in time.

Address translation mechanisms which employ both segmentation and paging are well known in the art. There are two common subcategories within this area of virtual address translation schemes: address translation in which paging is an integral part of the segmentation mechanism; and, address translation in which paging is independent from segmentation.

In prior art address translation mechanisms where paging is an integral part of the segmentation mechanism, the page translation can proceed in parallel with the segment translation since segments must start at page boundaries and are fixed at an integer number of pages. The segment number typically identifies a specific page table and the segment offset identifies a page number (through the page table) and an offset within that page. While this mechanism has the advantage of speed (since the steps can proceed in parallel) it is not flexible (each segment must start at a fixed page boundary) and is not optimal from a space perspective (*e.g.* an integer number of pages must be used, even when the segment may only spill over to a fraction of another page).

In prior art address translation mechanisms where paging is independent from segmentation, page translation generally cannot proceed until an intermediate, or linear, address is first calculated by the segmentation mechanism. The resultant linear address is then mapped onto a specific page number and an offset within the page by the paging mechanism. The page number identifies a page frame through a page table, and the offset identifies the offset within that page. In such mechanisms, multiple segments can be allocated into a single page, a single segment can comprise multiple pages, or a combination of the above, since segments are allowed to start on any byte boundary, and have any byte length. Thus, in these systems, while there is flexibility in terms of the segment/page relationship, this flexibility comes at a cost of decreased address translation speed.

Certain prior art mechanisms where segmentation is independent from paging allow for optional paging. The segmentation step is always applied, but the paging step is either performed or not performed as selected by the operating system.

These mechanisms typically allow for backward compatibility with systems in which segmentation was present, but paging was not included.

Typical of the prior art known to the Applicant in which paging is integral to segmentation is the Multics virtual memory, developed by Honeywell and described by the book, "The Multics System", by Elliott Organick. Typical of the prior art known to the Applicant in which optional paging is independent from segmentation is that described in U.S. Pat. No. 5,321,836 assigned to the Intel Corporation, and that described in the Honeywell DPS-8 Assembly Instructions Manual. Furthermore, U.S. Patent 4,084,225 assigned to the Sperry Rand Corporation contains a detailed discussion of general segmentation and paging techniques, and presents a detailed overview of the problems of virtual address translation.

Accordingly, a key limitation of the above prior art methods and implementations where segmentation is independent from paging is that the linear address must be fully calculated by the segmentation mechanism each time before the page translation can take place for each new virtual address. Only subsequent to the linear address calculation, can page translation take place. In high performance computer systems computer systems, this typically takes two full or more machine cycles and is performed on each memory reference. This additional overhead often can reduce the overall performance of the system significantly.

## SUMMARY OF THE INVENTION

An object of the present invention, therefore, is to provide the speed performance advantages of integral segmentation and paging and, at the same time, provide the space compaction and compatibility advantages of separate segmentation and paging.

A further object of the present invention is to provide a virtual address translation mechanism which architecturally provides for accelerating references to main memory in a computer system which employs segmentation, or which employs both segmentation and optional paging.

Another object of the present invention is to provide additional caching of page information in a virtual address translation scheme.

An further object of the present invention is to provide a virtual address translation mechanism which reduces the number of references required to ensure memory access.

According to the present invention, a segmentation unit converts a virtual address consisting of a segment identifier and a segment offset into a linear address. The segmentation unit includes a segment descriptor memory, which is selectable by the segment identifier. The entry pointed to by the segment identifier contains linear address information relating to the specific segment (i.e., linear address information describing the base of the segment referred to by the segment identifier, linear address information describing the limit of the segment referred to by the segment identifier, etc.) as well as physical address information pertaining to the segment - such as the page base of at least one of the pages represented by said segment.

In the above embodiment, unlike prior art systems, both segmentation and paging information are kept in the segmentation unit portion of the address translation system. The caching of this page information in the segmentation unit permits the address translation process to occur at much higher speed than in prior art systems, since the physical address information can be generated without having to perform a linear to physical address mapping in a separate paging unit.

The page base information stored in the segmentation unit is derived from the page frame known from the immediately prior in time address translation on a segment-by-segment basis. In order to complete the full physical address translation (*i.e.*, a page frame number and page offset), the segmentation unit combines the page frame from the segment descriptor memory with the page offset field, and may store this result in a segmentation unit memory, which can be a memory table, or a register, or alternatively, it may generate the full physical address on demand.

This fast physical address generated by the segmentation unit based on the virtual address and prior page information can be used by a bus interface to access a

physical location in the computer memory subsystem, even before the paging unit has completed its translation of the linear address into a page frame and page offset. Thus, fewer steps and references are required to create a memory access. Consequently, the address translation step occurs significantly faster. Since address translation occurs in a predominant number of instructions, overall system performance is improved.

The memory access is permitted to proceed to completion unless a comparison of the physical address information generated by the paging unit with the fast physical address generated by the segmentation unit shows that the page frame information of the segmentation unit is incorrect.

In alternative embodiments, the segmentation unit either generates the page offset by itself (by adding the lower portion of the segment offset and the segment base address) or receives it directly from the paging unit.

In further alternate embodiments, the incoming segment offset portion of the virtual address may be presented to the segmentation unit as components. The segmentation unit then combines these components in a typical base-plus-offset step using a conventional multiple input (typically 3-input ) adder well known in the prior art.

As shown herein in the described invention, the segment descriptor memory may be a single register, a plurality of registers, a cache, or a combination of cache and register configurations.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a typical prior art virtual address translation mechanism using segmentation and independent paging.

Figure 2A is a detailed diagram of a typical segment descriptor register of the prior art.

Figure 2B is a detailed diagram of an embodiment of the present invention, including a portion of a segment descriptor memory used for storing physical address information;

Figure 3A is a block diagram of an embodiment of the present invention employing segmentation and optional paging, and showing the overall structure and data paths used when paging is disabled during an address translation;

Figure 3B is a block diagram of the embodiment of Figure 3A showing the overall structure and data paths used when paging is enabled during an address translation;

Figure 3C is a block diagram of another embodiment of the present invention, showing an alternative circuit for generating the fast physical address information.

## DETAILED DESCRIPTION OF THE INVENTION

### *General Discussion of Paging & Segmentation*

The present invention provides for improved virtual address translation in a computer system. The preferred embodiment employs the invention in a single-chip microprocessor system, however, it is well understood that such a virtual address translation system could be implemented in multiple die and chip configurations without departing from the spirit or claims of the present invention.

Before embarking on a specific discussion of the present invention, however, a brief explanation of the general principles of segmentation and paging follows in order to provide additional background information, and so that the teachings of the present invention may be understood in a proper context.

Referring to FIG.1, a typical prior art virtual address translation mechanism 100 using both segmentation and, optionally, paging in a computer system is shown. As described in this figure, a data path within the microprocessor transmits a virtual address 101, consisting of segment identifier 101a and a segment offset 101b, to segmentation unit 130. Segments are defined by segment descriptor entries in at least one segment descriptor table or segment descriptor segment (not shown). Segment descriptor tables are created and managed by the operating system of the computer system, and are usually located in the memory subsystem. Segment descriptor entries are utilized in the CPU of the computer system by loading them

into segment descriptor register 190 or a segment descriptor cache (not shown); the segment descriptor register/cache is usually internal to the CPU, and thus more quickly accessible by the translation unit.

In the paging unit 150, pages are defined by a page table or multiple page tables (not shown), also created and managed by the operating system; again, these tables are also typically located in a memory subsystem. All or a portion of each page table can be loaded into a page cache 107 (within the CPU, sometimes called a translation look-aside buffer) to accelerate page references.

In operation, the segmentation unit 130 first translates a virtual address to a linear address and then (except in the case when optional paging is disabled) paging unit 150 translates the linear address into a real (or physical) memory address.

Typically (as in an x86 microprocessor) the segmentation unit translates a 48-bit virtual address 101 consisting of a 16-bit segment identifier (<s>) 101a and a 32-bit displacement within that segment (<d>) 101b to a 32-bit linear (intermediate) address 106. The 16-bit segment identifier 101a uniquely identifies a specific segment; this identifier is used to access an entry in a segment descriptor table (not shown). In the prior art, this segment descriptor entry contains a base address of the segment 191, the limit of the segment 192, and other attribute information described further below. The segment descriptor entry is usually loaded into a segment descriptor register 190.

Using adder 105, the segmentation unit adds the segment base 191 of the segment to the 32-bit segment offset 101b in the virtual address to obtain a 32-bit linear address. The 32-bit segment offset 101b in the virtual address is also compared against the segment limit 192, and the type of the access is checked against the segment attributes. A fault is generated and the addressing process is aborted if the 32-bit segment offset is outside the segment limit, or if the type of the access is not allowed by the segment attributes.

The resulting linear address 106 can be treated as an offset within a linear address space; and in the commonly implemented schemes of the prior art, these offsets are frequently byte offsets. When optional paging is disabled, the linear

address 106 is exactly the real or physical memory address 108. When optional paging is enabled, the linear address is treated as a 2- or 3-tuple depending on whether the paging unit 150 utilizes one or two level page tables.

In the 2-tuple case shown in FIG.1, which represents single level paging, the linear address, <<p>,<pd>> is divided into a page number field <p> 106a, and a page displacement (page offset) field within that page (<pd>) 106b. In the 3-tuple case (not shown) <<dp>,<p>,<pd>>, the linear address is divided into a page directory field (<dp>), a page number field <p> and a page displacement field <pd>. The page directory field indexes a page directory to locate a page table (not shown). The page number field indexes a page table to locate the page frame in real memory corresponding to the page number, and the page displacement field locates a byte within the selected page frame. Thus, paging unit 150 translates the 32-bit linear address 106 from the segmentation unit 130 to a 32-bit real (physical) address 108 using one or two level page tables using techniques which are well known in the art.

In all of the above prior art embodiments where segmentation is independent from paging, the segment descriptor table or tables of the virtual address translator are physically and logically separate from the page tables used to perform the described page translation. There is no paging information in the segment descriptor tables and, conversely, there is no segmentation information in the page tables.

This can be seen in FIG. 2A. In this figure, a typical prior art segment descriptor entry 200, is shown as it is typically used in a segment descriptor table or segment descriptor register associated with a segmentation unit. As can be seen there, segment descriptor 200 includes information on the segment base 201, the segment limit 202, whether the segment is present (P) 203 in memory, the descriptor privilege level (DPL) 204, whether the segment belongs to a user or to the system (S) 205, and the segment type 206 (code, data, stack, etc.)

For additional discussions pertaining to the prior art in segmentation, paging, segment descriptor tables, and page tables, the reader is directed to the references U.S. Patent Nos. 5,408,626, 5,321,836, 4,084,225, which are expressly incorporated by reference herein.

*Improved Segmentation Unit Using Paging Information*

As shown in the immediate prior art, the paging and segmentation units (circuits) are completely separate and independent. Since the two units perform their translation sequentially, that is, the segment translation must precede the page translation to generate the linear address, high performance computer systems, such as those employing superscalar and superpipelined techniques, can suffer performance penalties. In some cases, it is even likely that the virtual address translation could fall into the systems' "critical path". The "critical path" is a well-known characteristic of a computer system and is typically considered as the longest (in terms of gate delay) path required to complete a primitive operation of the system.

Accordingly, if the virtual address translation is in the critical path, the delays associated with this translation could be significant in overall system performance. With the recognition of this consideration, the present invention includes page information in the segment translation process. The present invention recognizes the potential performance penalty of the prior art and alleviates it by storing paging information in the segmentation unit obtained from a paging unit in previous linear-to-real address translations.

As can be seen in FIG. 2B, the present invention extends the segment descriptor entries of the prior art with a segment entry 290 having two additional fields: a LAST PAGE FRAME field 297 and a VALID field 298. The LAST PAGE FRAME field 297 is used to hold the high-order 20 bits (*i.e.*: the page frame) of the real (physical) memory address of the last physical address generated using the specified segment identifier. The VALID field 298 is a 1-bit field, and indicates whether or not the LAST PAGE FRAME field 297 is valid. The remaining fields

291-296 perform the same function as comparable fields 201-206 respectively described above in connection with FIG. 2A.

Segment descriptor tables (not shown) can be located in a memory subsystem, using any of the techniques well-known in the art. As is also known in the art, it is possible to speed up address translation within the segmentation unit by using a small cache, such as one or more registers, or associative memory. The present invention makes use of such a cache to store segment entries 290 shown above. Unlike the prior art, however, the segment entries 290 of the present invention each contain information describing recent physical address information for the specified segment. Accordingly, this information can be used by a circuit portion of the segmentation unit to generate a new physical address without going through the linear to physical mapping process typically associated with a paging unit.

While in some instances the physical address information may change between two time-sequential virtual addresses to the same segment (and thus, a complete translation is required by both the segmentation and paging units), in the majority of cases the page frame information will remain the same. Thus, the present invention affords a significant speed advantage over the prior art, because in the majority of cases a complete virtual-linear-physical address translation is not required before a memory access is generated.

*Embodiment With Segmentation & Optional Paging/Paging Disabled*

Referring to FIG. 3A, the advantage of using this new information in segment entry 290 in a segmentation unit or segmentation circuit is apparent from a review of the operation of an address translation. In this figure, a paging unit (or paging circuit) is disabled, as for example might occur only when a processor is used in a real mode of operation, rather than a protected mode of operation.

In a preferred embodiment, the present invention employs a segment descriptor memory comprising at least one, and preferably many, segment descriptor registers 390, which are identical in every respect to the segment descriptor register described above in connection with Fig.2B. These segment descriptor registers are

loaded from conventional segment descriptor tables or segment descriptor segments which are well known in the art. Each segment descriptor register 390 is loaded by the CPU before it can be used to reference physical memory. Segment descriptor register 390 can be loaded by the operating system or can be loaded by application programs. Certain instructions of the CPU are dedicated to loading segment descriptor registers, for example, the "LDS" instruction, "Load Pointer to DS Register". Loading by the operating system, or execution of instructions of this type, causes a base 391, limit 392, descriptor privilege level 394, system/user indicator 395, and type 396 to be loaded from segment tables or segment descriptor segments as in the prior art. The three remaining fields are present 393, LAST PAGE FRAME 397 and VALID 398. When a segment descriptor register 390 is loaded, present 393 is set to 1, indicating that the segment descriptor register 390 contents are present; the valid field 398 is set to 0, indicating that the last page frame number field 397 is not valid; and the LAST PAGE FRAME field 397 is not set, or may be set to 0.

After the loading of a segment descriptor register 390, instructions of the CPU may make references to virtual memory; if a segment descriptor register is referenced before it is loaded, as indicated by present field 393 set to 0, a fault occurs and the reference to the segment descriptor register is aborted.

As explained above, the CPU makes references to virtual memory by specifying a 48-bit virtual address, consisting of a 16-bit segment identifier 301a and a 32-bit segment offset 301b. A data path within the CPU transmits virtual address 301 to the address translation mechanism 300.

Segment descriptor memory 390 is indexed by segment number, so each entry in this memory containing data characteristics (*i.e.*, base, access rights, limit) of a specific segment is selectable by the segment identifier from the virtual address. Assuming this is the first reference to physical memory specifying a newly loaded segment descriptor register, since the VALID bit 398 is set to false, a prior art virtual address translation takes place. This involves, among other things, as explained earlier, various validity checks (including checking attributes 394-396,

segment limit checking using comparator 302 and potentially others), and using adder 305 to add the segment descriptor's base address 391 to the segment offset 301b to calculate a linear address 306.

While the implementation in the embodiment of Fig. 3A shows the addition of the base address 391 to the segment offset 301b using adder 305 to generate the linear address 306, it will be understood by those skilled in the art that this specific implementation of the virtual to linear address translation is not the only implementation of the present invention. In other implementations, the segment offset 301b might consist of one or more separate components. Different combinations of one or more of these components might be combined using well known techniques to form a linear address, such as one utilizing a three-input adder. The use of these components is discussed, for example, in U.S. Patent No. 5,408,626, and that description is incorporated by reference herein.

As is well known, in this embodiment where paging is disabled, linear address 306 is also a physical address which can be used as the physical address 308. Memory access control operations are not shown explicitly since they are only ancillary to the present invention, and are well described in the prior art. In general, however, a bus interface unit 380 is typically responsible for interactions with the real (physical) memory subsystem. The memory subsystem of a computer system employing the present invention preferably has timing and address and data bus transaction details which are desirably isolated from the CPU and address translation mechanism. The bus interface unit 380 is responsible for this isolation, and can be one of many conventional bus interface units of the prior art.

In the present invention, bus interface unit 380 receives the real memory address 308 from address translation mechanism 300 and coordinates with the real memory subsystem to provide data, in the case of a memory read request, or to store data, in the case of a memory write request. The real memory subsystem may comprise a hierarchy of real memory devices, such as a combination of data caches and dynamic RAM, and may have timing dependencies and characteristics which are

isolated from the CPU and address translation mechanism 300 of the computer system by the bus interface unit 380.

Simultaneous with the first memory reference using the calculated physical address 308, the LAST PAGE FRAME field of the selected segment descriptor register 390 is loaded with the high-order 20 bits of the physical address, i.e.: the physical page frame, and the VALID bit is set to indicate a valid state. This paging information will now be used in a next virtual address translation.

Accordingly, when a next, new virtual address 301 is to be translated, the entry selected from segment descriptor memory 390 will likely contain the correct physical frame page number (in the LAST PAGE FRAME field 397). Thus, in most cases, the base physical address in memory for the next, new referenced virtual address will also be known from a previous translation.

The first step of the virtual address translation, therefore, is to determine if a FAST PHYSICAL ADDRESS 303 can be used to begin a fast physical memory reference. Adder 309, a 12-bit adder, adds the low-order 12 bits of the segment offset 301b of virtual address 301 to the low-order 12-bits of base 391 of the segment entry in segment descriptor register 390 referenced by the segment identifier 301a. This addition results in a page offset 303b. In parallel with adder 309, 32-bit adder 305 begins a full 32-bit add of segment base 391 and segment offset 301b, to begin producing the linear address; however, this full 32-bit add will obviously require more time. In the preferred embodiment, adder 309 is a separate 12-bit adder; however, it should be noted that adder 309 also could be implemented as the low order 12-bits of 32-bit adder 305.

Simultaneous with the beginning of these two operations, VALID bit 398 is inspected. If VALID bit 398 is set to 1, as soon as 12-bit adder 309 has completed, 20-bit LAST PAGE FRAME 397 is concatenated with the result of adder 309 to produce FAST PHYSICAL ADDRESS 303, consisting of a page frame number 303a, and page offset 303b. FAST PHYSICAL ADDRESS 303 then can be used to tentatively begin a reference to the physical memory. It should be understood that

the FAST PHYSICAL ADDRESS 303 transmitted to bus interface unit 380 could also be stored in a register or other suitable memory storage within the CPU.

In parallel with the fast memory reference, limit field 392 is compared to the segment offset 301b of the virtual address by comparator 302. If the offset in the virtual address is greater than the limit, a limit fault is generated, and virtual address translation is aborted.

Also in parallel with the fast memory reference, adder 305 completes the addition of base 391 to the segment offset field 301b of virtual address to produce linear address (in this case physical address also) 306. When this calculation is completed, the page frame number 308a of physical address 308 is compared to LAST PAGE FRAME 397 by Not Equal Comparator 304. If page frame 308a is unequal to the LAST PAGE FRAME 397, or if 12-bit Adder 309 overflowed (as indicated by a logic "1" at OR gate 310), the fast memory reference is canceled, and the linear address 306, which is equal to the physical address 308, is used to begin a normal memory reference. If page frame 308a is equal to LAST PAGE FRAME 397, and 12-bit Adder 309 did not overflow (the combination indicated by a logic "0" at the output of OR gate 310), the fast memory reference is allowed to fully proceed to completion.

After any fast memory reference which is cancelled by the CANCEL FAST PHYSICAL ADDRESS signal output of OR gate 310, page frame 308a is loaded into the LAST PAGE FRAME 397 in the segment descriptor memory 390 for subsequent memory references.

Depending on the particular design desired, it should also be noted that writes to the memory, or reads which cause faults using FAST PHYSICAL ADDRESS 303 may be pended since the FAST PHYSICAL ADDRESS 303 may prove to be invalid.

Accordingly, it can be seen that the parallel physical address calculation undertaken by the improved segmentation unit of the present invention generates a faster physical memory access than possible with prior art systems.

*Embodiment With Segmentation & Paging/Paging Enabled*

The present invention can also be used with address translation units using paging enabled, as can be seen in the embodiments of FIGs. 3B and 3C.

In the embodiment of FIG. 3B, the same segmentation unit structure 300 as that shown in FIG. 3A is used, and the operation of segmentation unit 300 is identical to that already explained above. As before, segment descriptor memory (registers) 390 are loaded from conventional segment descriptor tables or segment descriptor segments, using one or more of the procedures described above. First, the base 391 limit 392 descriptor privilege level 394, system/user indicator 395, and type 396 are loaded from segment tables or segment descriptor segments as explained earlier. When segment descriptor register 390 is loaded, present 393 is set to 1, indicating that the segment descriptor register 390 contents are present; the valid field 398 is set to 0, indicating that the last page frame number field 397 is not valid; and the LAST PAGE FRAME field 397 is not set, or may be set to 0.

As explained above, after the loading of a segment descriptor register 390, instructions of the CPU may make references to virtual memory; if a segment descriptor register is referenced before it is loaded, as indicated by present field 393 set to 0, a fault occurs and the reference to the segment descriptor register is aborted.

As further explained above, the 48 bit virtual address 301 (consisting of a 16 bit segment identifier 301a and a 32 bit segment offset 301b) is transmitted by a data path to segmentation unit 300, and an index into segment descriptor memory 390 is performed to locate the specific segment descriptor for the segment pointed to by segment identifier 301a. Assuming this is the first reference to physical memory specifying a newly loaded segment descriptor register, since the VALID bit is set to false, a prior art virtual address translation takes place. This involves, among other things, as explained earlier, various validity checks (including checking attributes 394-396, segment limit checking using comparator 302 and potentially others), and using adder 305 to add the segment descriptor's base address 391 to the segment offset 301b to calculate a linear address 306.

As is well known, in this configuration where paging is enabled, linear address 306 must undergo a further translation by paging unit 350 to obtain the physical address 308 in the memory subsystem. In the preferred embodiment of the invention, looking first at FIG.3B, the output of adder 305 will be a 32-bit linear address, corresponding to a 20-bit page number 306a and a 12-bit page offset 306b. Typically, the page number 306a is then indexed into a page descriptor table (not shown) to locate the appropriate page frame base physical address in memory. These page descriptor tables are set up by the operating system of the CPU using methods and structures well known in the art, and they contain, among other things, the base physical address of each page frame, access attributes, etc.

However, in most systems, including the present invention, a page cache 307 is used in order to hold the physical base addresses of the most recently used page frames. This cache can take the form of a table, associative cache, or other suitable high speed structure well known in the art. Thus, page number 306a is used to access page data (including physical base addresses for page frames) in an entry in page cache 307.

If page cache 307 hits, two things happen: first, a 20-bit PAGE FRAME 307a (the page frame in physical memory) replaces the high-order 20 bits (page number 306a) of the linear address 306, and, when concatenated with the page offset 306b results in a real (physical) address 308, which is used to perform a memory access through bus interface unit 380 along the lines explained above. Second, newly generated page frame 308a is also stored in segment descriptor memory 390 in the selected LAST PAGE FRAME field 397 to be used for a fast access in the next address translation. When LAST PAGE FRAME field 397 is stored, selected VALID bit 398 is set to 1 to indicate that LAST PAGE FRAME 397 is valid for use.

In the event of a page cache miss, the appropriate page frame number 308a is located (using standard, well-known techniques) to generate physical address 308, and is also loaded into segment descriptor memory 390 in the selected LAST PAGE FRAME field 397. The selected VALID bit 398 is also set to indicate a valid state.

Thus, there is paging information in the segmentation unit that will now be used in the next virtual address translation.

When a next, new virtual address 301 is to be translated, the segment identifier 301a will likely be the same as that of a previously translated virtual address, and the entry selected from segment descriptor memory 390 will also likely contain the correct physical frame (in LAST PAGE FRAME field 397) from the previous translation. As with the above embodiment, one or more registers, or a cache may be used for the segment descriptor memory 390.

The first step then determines if a FAST PHYSICAL ADDRESS 303 can be used to begin a fast physical memory reference. Adder 309, a 12-bit adder, adds the low-order 12 bits of the segment offset 301b of virtual address 301 to the low-order 12-bits of base 391 of the segment entry in segment descriptor register 390 referenced by the segment identifier 301a. This addition results in a page offset 303b. In parallel with adder 309, 32-bit adder 305 begins a full 32-bit add of segment base 301 and segment offset 301b, to begin producing the linear address; however, this full 32-bit add will obviously require more time. In the preferred embodiment, adder 309 is a separate 12-bit adder; however, it should be noted that adder 309 also could be implemented as the low order 12-bits of 32-bit adder 305.

Simultaneous with these beginning of these two operations, VALID bit 398 is inspected. If VALID bit 398 is set to 1, as soon as 12-bit adder 309 has completed, 20-bit LAST PAGE FRAME 397 is concatenated with the result of adder 309 to produce FAST PHYSICAL ADDRESS 303, consisting of a page frame number 303a, and page offset 303b. FAST PHYSICAL ADDRESS 303 then can be used to tentatively begin a reference to the physical memory. Again, it should be understood that the FAST PHYSICAL ADDRESS 303 transmitted to bus interface unit 380 could also be stored in a register or other suitable memory storage within the CPU.

As before, limit field 302 is compared to the segment offset 301b of the virtual address by comparator 302. If the offset in the virtual address is greater than the limit, a limit fault is generated, and virtual address translation is aborted.

This new virtual address is also translated by paging unit 350 in the same manner as was done for the previous virtual address. If page cache 307 hits based on the page number 306a, two things happen: first, a 20-bit PAGE FRAME 307a (the page frame in physical memory) replaces the high-order 20 bits (page number 306a) of the linear address 306, and, when concatenated with the page offset 306b results in a physical address 308. This real address may or may not be used, depending on the result of the following: in parallel with the aforementioned concatenation, the PAGE FRAME 307a, is compared to LAST PAGE FRAME 397 from the segment descriptor memory 390 by Not Equal Comparator 304. The result of Not Equal Comparator (that is, the Not Equal condition) is logically ORed with the overflow of 12-bit adder 309 by OR gate 310. If the output of OR gate 310 is true (i.e. CANCEL FAST PHYSICAL ADDRESS is equal to binary one), or if PAGE CACHE 307 indicates a miss condition, the fast memory reference previously begun is canceled, since the real memory reference started is an invalid reference. Otherwise, the fast memory reference started is allowed to fully proceed to completion, since it is a valid real memory reference.

If CANCEL FAST PHYSICAL ADDRESS is logical true, it can be true for one of two, or both reasons. In the case that Or gate 310 is true, but page cache 307 indicates a hit condition, physical address 308 is instead used to start a normal memory reference. This situation is indicative of a situation where LAST PAGE FRAME 397 is different from the page frame 308a of the current reference.

In the case that page cache 307 did not indicate a hit, a page table reference through the page descriptor table is required and virtual address translation proceeds as in the prior art. The page frame 308a information is again stored in the LAST PAGE FRAME field 397 in the segment descriptor memory 390 for the next translation.

Also, after any fast memory reference which is canceled by the CANCEL FAST PHYSICAL ADDRESS signal output of OR gate 310, page frame number 308a is loaded into the LAST PAGE FRAME 397 in the segment descriptor memory 390 for subsequent memory references.

Depending on the particular design desired, it should also be noted that in this embodiment also, writes to the memory, or reads which cause faults using FAST PHYSICAL ADDRESS 303 may be pended since the FAST PHYSICAL ADDRESS 303 may prove to be invalid.

The alternative embodiment shown in FIG. 3C is identical in structure and operation to the embodiment of FIG. 3B, with the exception that the 12-bit Adder 309 is not employed. In this embodiment, the segmentation unit 330 does not create the lower portion (page offset 303a) of the fast physical address in this manner. Instead, the page offset 306a resulting from 32-bit adder 305 is used.

It can be seen that the present invention has particular relevance to computers using sequential type of segmentation and paging translation, such as the X86 family of processors produced by the Intel Corporation (including the Intel 80386, Intel 80486 and the Intel Pentium Processor), other X86 processors manufactured by the NexGen Corporation, Advanced Micro Devices, Texas Instruments, International Business Machines, Cyrix Corporation, and certain prior art computers made by Honeywell. These processors are provided by way of example, only, and it will be understood by those skilled in the art that the present invention has special applicability to any computer system where software executing on the processors is characterized by dynamic execution of instructions in programs in such a way that the virtual addresses are generally logically and physically located near previous virtual addresses.

The present invention recognizes this characteristic, employing acceleration techniques for translating virtual to real addresses. In particular, the present invention utilizes any of the commonly known storage structures (specific examples include high speed registers and/or caches) to store previous address translation information, and to make this previous address translation information available to the system whenever the next subsequent reference relies on the same information. In this way, the system can utilize the previously stored information from the high speed storage to begin real memory references, rather than be forced to execute a

more time consuming translation of this same information, as was typically done in the prior art.

As will be apparent to those skilled in the art, there are other specific circuits and structures beyond and/or in addition to those explicitly described herein which will serve to implement the translation mechanism of the present invention. Finally, although the above description enables the specific embodiment described herein, these specifics are not intended to restrict the invention, which should only be limited as defined by the following claims.

I claim:

1. A circuit for storing address translation information, said circuit comprising:
   a) a data path for receiving a virtual address, said virtual address including a segment identifier and a segment offset; and
   b) a segment descriptor memory coupled to said data path and selectable by said segment identifier, said memory capable of storing at least the following:
      I) linear address information describing the base of the segment,
      ii) linear address information describing the limit of the segment, and
      iii) a page frame describing at least a portion of a physical address of said segment.

2. The circuit of claim 1, wherein the segment descriptor memory is one or more registers.

3. The circuit of claim 1, wherein the segment descriptor memory is a cache.

4. The circuit of claim 1, further including a physical address register coupled to the segment descriptor memory for storing a physical address, said physical address

being comprised of the page frame from said segment descriptor memory and a page offset.

5. The circuit of claim 4, wherein the physical address stored in the physical address register is used to perform a memory access.

6. The circuit of claim 1, wherein the segment descriptor memory is further capable of storing:

(iv) information describing whether said page frame can be used for an address translation.

7. In a system having both a segmentation unit and a paging unit for translating physical addresses from virtual addresses, said virtual addresses including a segment identifier and segment offset, the improvement wherein:

the segmentation unit includes a segment descriptor memory selectable by said segment identifier, said memory capable of storing at least the following:

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment, and

iii) physical address information relating to said segment.

8. A circuit for storing address translation information in a computer system, said computer using both segmentation and paging to translate a virtual address having a segment identifier and segment offset into a physical address, said circuit comprising:

a segment descriptor memory selectable by said segment identifier, said memory capable of storing at least the following:

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment, and

iii) page frame describing at least a portion of a physical address of said segment.

9. An address translation system for translating a virtual address having a segment identifier and an offset field into a physical address, said address translation system being used in a computer system and comprising:

(a) a segmentation unit for generating linear address information based on the virtual address information, and for storing first physical address information having a first physical page frame and a first physical page offset;

(b) a paging unit coupled to the segmentation unit for generating said first physical page frame and second physical address information having a second physical page frame and a second page offset; and

© said segmentation unit further including:

[1] a segment descriptor memory, selectable by said segment identifier of said virtual address, and capable of storing

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment,

iii) said first physical page frame;

[2] a limit comparator for comparing whether the offset field exceeds the limit of the segment;

wherein a system memory access can be made by said computer system based on said first physical address information.

10. The system of claim 9, wherein the segmentation unit further includes a physical address comparator for comparing the first physical page frame and the second physical page frame, and wherein the memory access is canceled only if the first physical page frame and second physical page frame are not equal.

11. The system of claim 9, wherein the segmentation unit further includes an adder, and the first physical page offset can be generated by adding a portion of the virtual address offset field and a portion of the segment base in the segment descriptor memory.

12. The system of claim 9, wherein the first physical page offset can be generated by adding the virtual address offset field and the segment base in the segment descriptor memory.

13. The system of claim 9, wherein the paging unit further includes a page cache for storing page frames of physical pages most recently used by the computer system.

14. The system of claim 9, wherein the segment descriptor memory comprises one at least a register and/or cache.

15. The system of claim 9, wherein the first physical page frame generated by the paging unit is stored in the segment descriptor memory of the segmentation unit and can be used in a subsequent virtual to physical address translation.

16. The system of claim 9, wherein the segment descriptor memory is further capable of storing:

(iv) information describing whether said page frame can be used for an address translation.

17. The system of claim 9, wherein the segmentation unit also uses either or both index and displacement information to generate the linear address information.

18. A system for address translation in a CPU comprising:

a) an instruction set employing virtual addresses for accessing data or instructions located at physical addresses in a memory subsystem, said virtual addresses containing a segment identifier field selecting a segment and a segment offset field selecting an offset within the selected segment;

b) a segmentation circuit for generating linear addresses based on the virtual addresses, and for storing physical address information, said segmentation circuit including at least one segment descriptor memory selectable by said segment identifier, said segment descriptor memory capable of storing:

(I) segment data describing a segment, said segment data including linear address information describing the base and limit of the segment in a linear address space;

(ii) a physical address information corresponding to a page frame;

c) a paging circuit for receiving the linear addresses and generating physical addresses, said paging circuit including a page cache, said page cache optionally storing page frame and page offset field information for said CPU; and

d) a bus interface, capable of coupling the segmentation and paging circuits of the CPU to the memory subsystem, and for performing memory accesses in response to physical address information from either of said segmentation and paging circuits.

19. The circuit of claim 18, wherein the segment descriptor memory is one or more registers, or a cache.

20. The system of claim 18, wherein said bus interface provides either separate address/data lines to said memory, or multiplexed address/data lines.

21. The system of claim 18, wherein the segmentation circuit further includes a physical address comparator for comparing the page frame in the segmentation circuit descriptor memory with the page frame in the page cache, and wherein the

memory access is canceled only if the page frame in the segmentation circuit segment descriptor memory is different from the page frame in the page cache.

22. The system of claim 18, wherein the segmentation circuit further includes an adder, and a page offset is generated by adding a portion of the segment offset field to a portion of the segment base in the segment descriptor memory.

23. The system of claim 18, wherein the page frame generated by the paging unit is stored in the segmentation circuit.

24. The system of claim 18 wherein the segment descriptor cache is further capable of storing:

(iv) information describing whether said physical address information can be used for an address translation.

25. The system of claim 18, wherein the segmentation circuit uses index and/or displacement information to generate the linear addresses.

26. An address translation system for translating a virtual address having a segment and an offset field into a physical address, said address translation system being used in a computer system and comprising:

a) a segmentation unit for generating linear address information based on the virtual address information, and for generating and storing a physical address including a page frame and a page offset, said segmentation unit further including:

b) [1] a segment descriptor memory, selectable by said segment identifier of said virtual address, and capable of storing

I) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment,

iii) said page frame;

[2] a physical address memory for storing said physical address including said page frame number and a page offset; and

wherein an access to a memory subsystem can be made by said computer system based on said physical address.

27. The system of claim 26, wherein the segmentation unit further includes a physical address comparator for comparing the page frame from said virtual address with a page frame corresponding to a different, later in time virtual address, and wherein the memory access is canceled only if the page frames of said different virtual addresses are not equal.

28. The system of claim 26, wherein the segmentation unit further includes an adder, and the page offset is generated by adding a portion of the virtual address offset field and a portion of the segment base in the segment descriptor memory, and then stored in a register corresponding to the first physical address memory.

29. The system of claim 26, wherein the segment descriptor memory includes at least a register and/or a cache.

30. The system of claim 26, further including a bus interface capable of coupling the segmentation unit to the memory subsystem, and for performing memory accesses in response to physical address information from said segmentation unit.

31. The system of claim 30, wherein said bus interface provides either separate address/data lines to said memory, or multiplexed address/data lines.

32. The system of claim 26, wherein the segment descriptor memory is further capable of storing:

(iv) information describing whether said page frame can be used for an address translation.

33. The system of claim 26, wherein the segmentation unit also uses either or both index and displacement information to generate the linear address information.

34. A method of calculating physical addresses from virtual addresses, said method comprising:

a) calculating a first physical address, having a first page frame field and a first page offset field, based on a virtual address;

b) storing said first page frame field of said first physical address;

c) calculating a second physical address based on a second virtual address including a second page frame field and a second page offset field;

d) generating a third physical address based on the first page frame field and the second page offset field;

e) generating a memory access request based on said third physical address.

35. The method of claim 34, further including a step: (f) canceling said access request to memory using said third physical address if the first page frame field is not equal to the second page frame field of said second physical address.

36. The method of claim 34, wherein the page frame fields most recently used by the computer system are stored.

37. The method of claim 34, further including a step: checking whether the first page frame field can be used for an address translation.

## ABSTRACT

An improved address translation method and mechanism for memory management in a computer system is disclosed. A segmentation mechanism employing segment registers maps virtual addresses into a linear address space. A paging mechanism optionally maps linear addresses into physical or real addresses. Independent protection of address spaces is provided at each level. Information about the state of real memory pages is kept in segment registers or a segment register cache potentially enabling real memory access to occur simultaneously with address calculation, thereby increasing performance of the computer system.

ORIGINAL

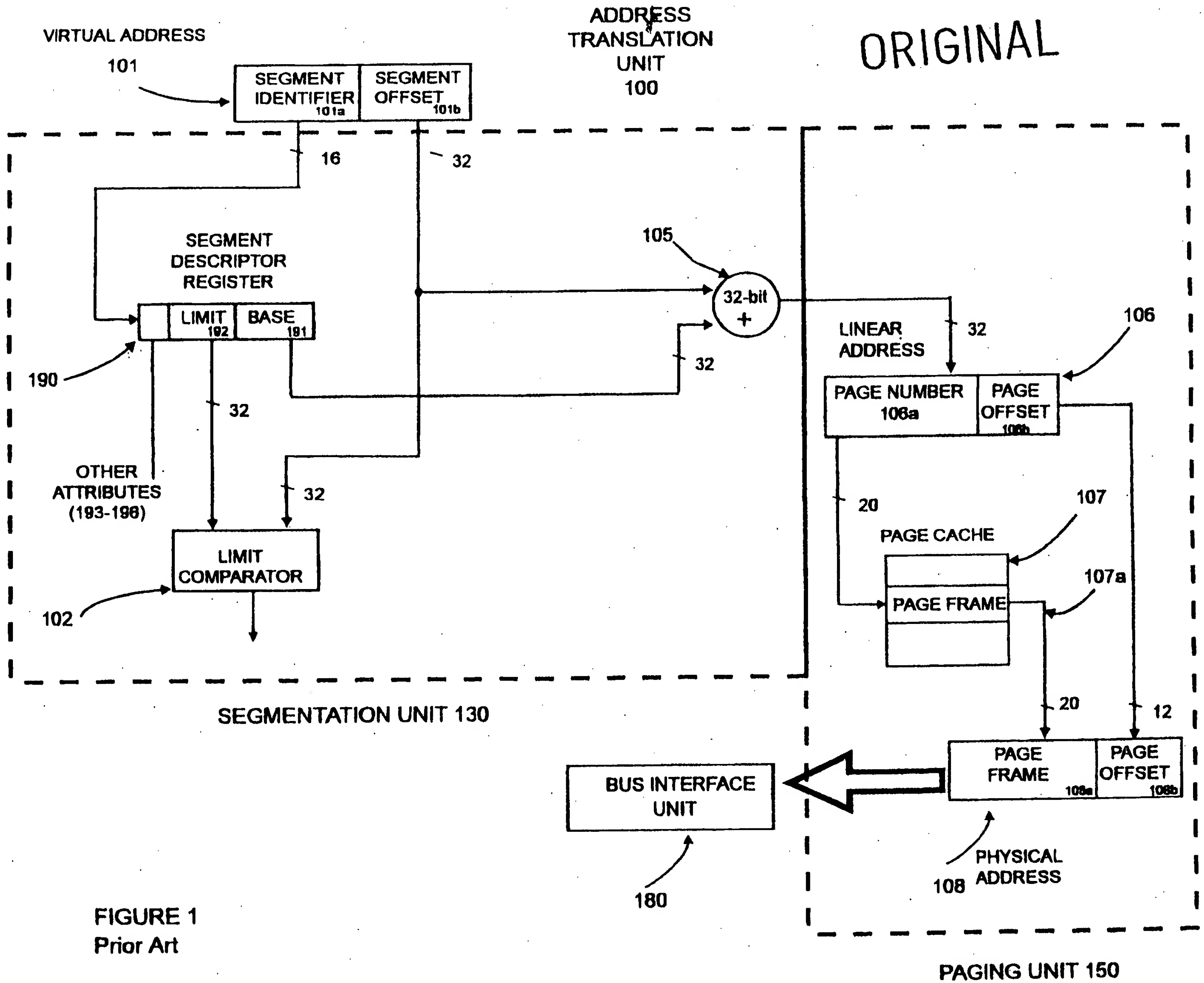
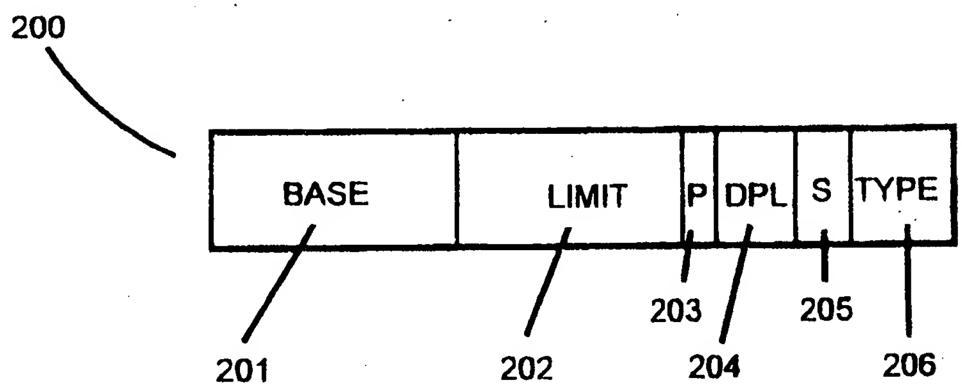



FIGURE 1
Prior Art

JNB
RAB 97-002

ORIGINAL




PRIOR ART SEGMENT DESCRIPTOR REGISTER

FIGURE 2A

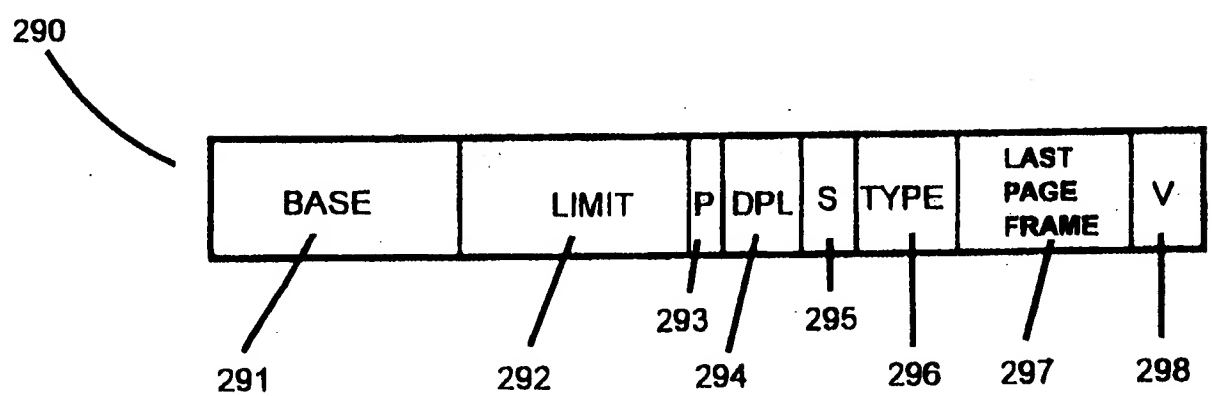



FIGURE 2B

SEGMENT DESCRIPTOR MEMORY

FIGURE 2

ORIGINAL

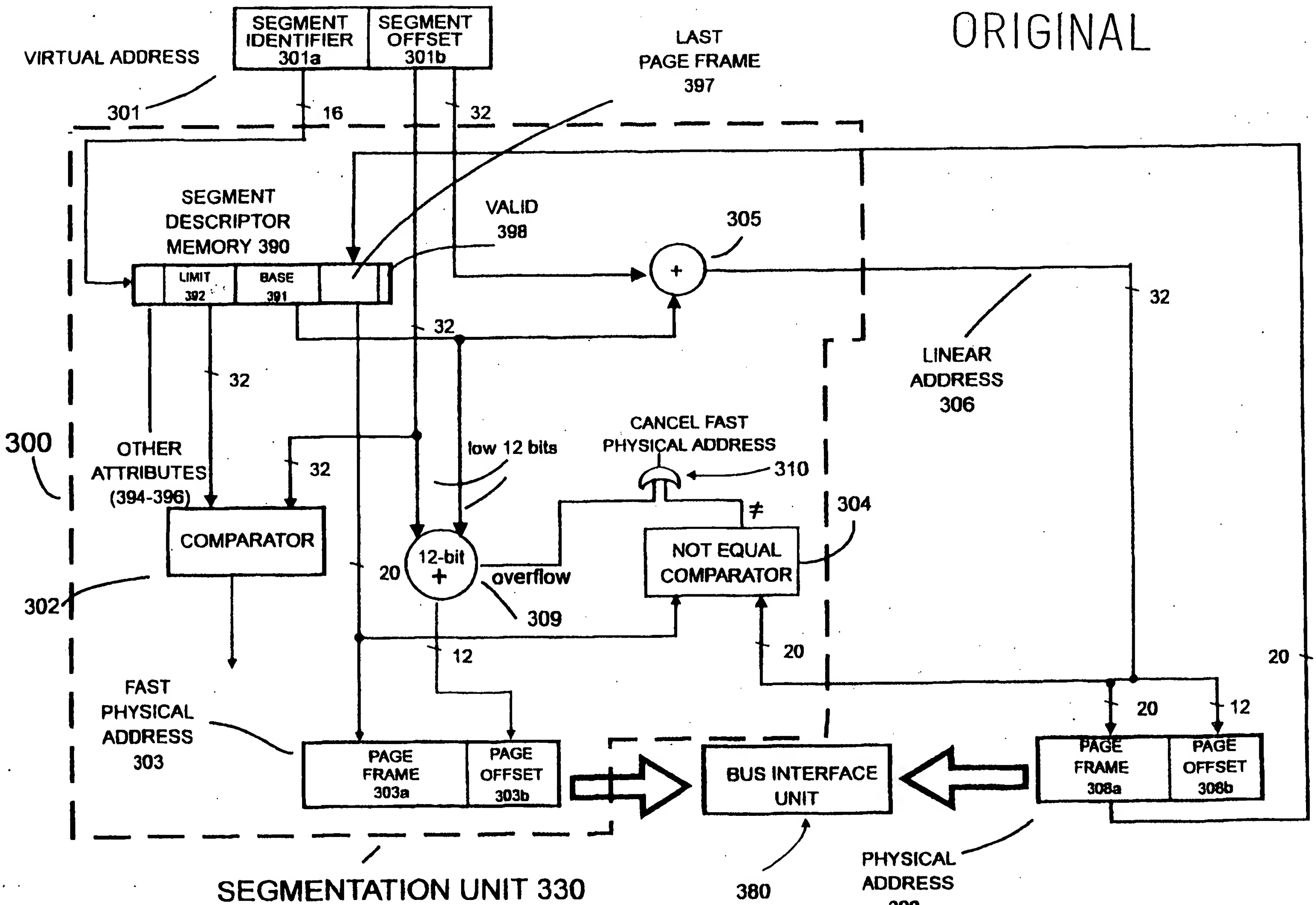



FIGURE 3A

ORIGINAL

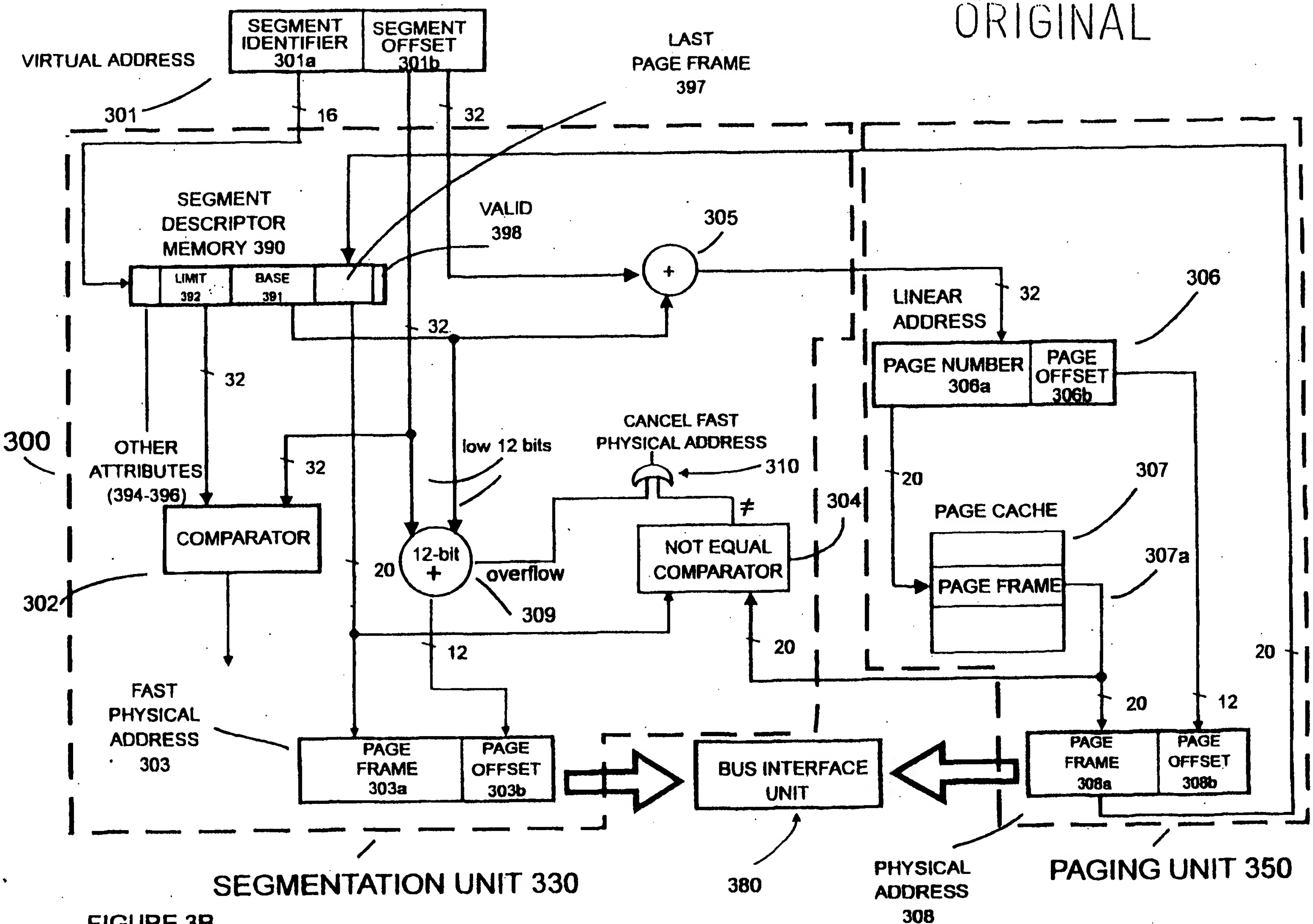



FIGURE 3B

ORIGINAL

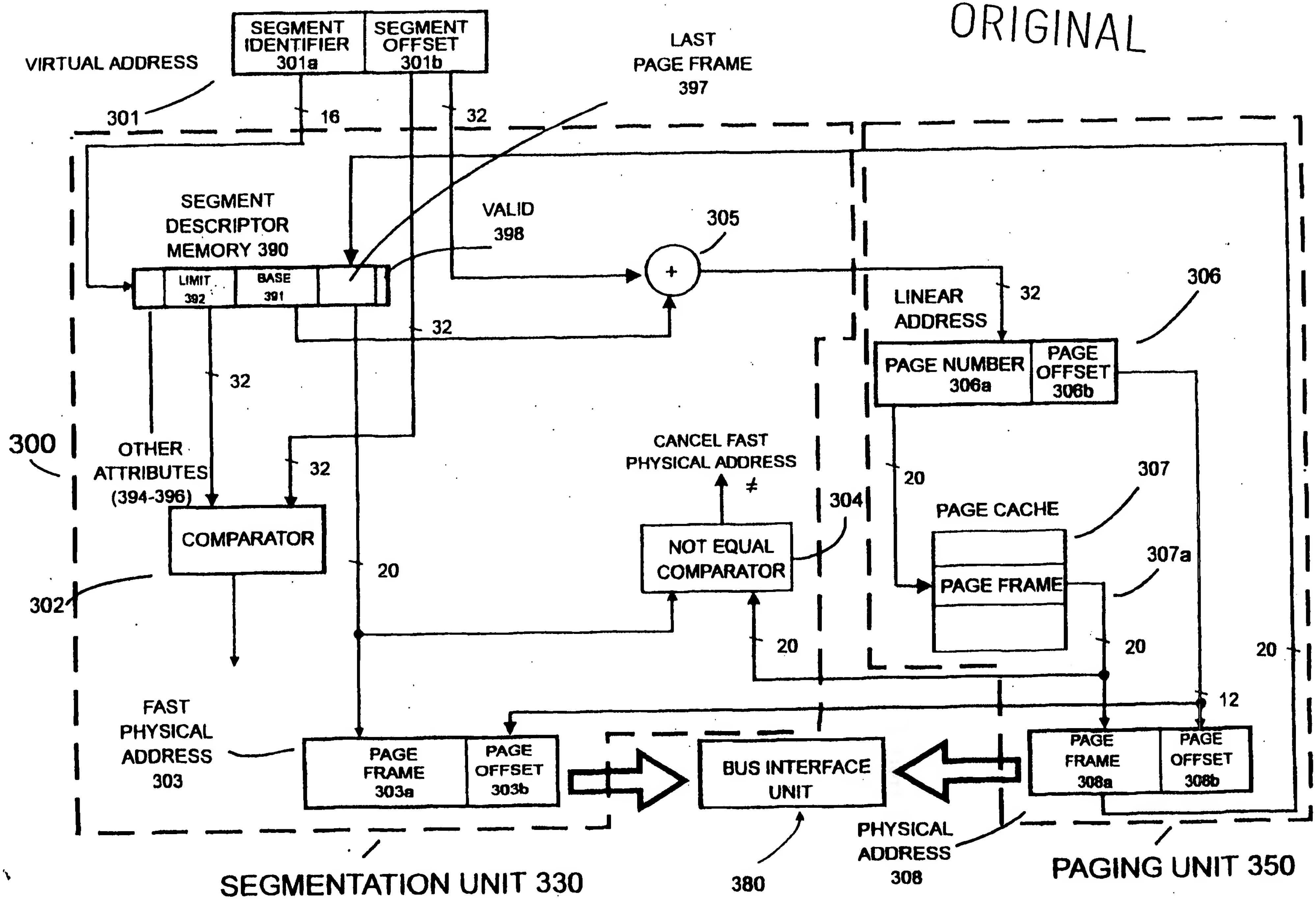



FIGURE 3C

JOI 6



RAB 97-002



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: *Belgard, R.*

Serial No.: *unknown*

Filed: *8/4/97 as continuation of parent application serial no. 08/458,479 filed 6/2/95*

For: *Address Translation Method and Mechanism Using Physical Address Information Including During A Segmentation Process*

Art Unit: *unknown*

Examiner: *unknown*

OIPE JC69 MAR 2 2 2000 PATENT & TRADEMARK OFFICE

---

PRELIMINARY AMENDMENT
FOR ACCOMPANYING RULE 1.60 CONTINUATION APPLICATION

Honorable Commissioner
of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

As part of the present filing, the Applicant respectfully submits the following to place this case in condition for allowance:

IN THE TITLE:

Please change the title to: FAST ADDRESS GENERATOR FOR REDUCING VIRTUAL-LINEAR-PHYSICAL ADDRESS CONVERSION TIME.

IN THE SPECIFICATION:

Please insert the following sentences before line 1: "This is a continuation of application serial no. 08/458,479 filed on June 2, 1995. The present application is also related to a further application filed concurrently herewith entitled COMPUTER ADDRESS TRANSLATION USING FAST ADDRESS GENERATOR DURING A SEGMENTATION OPERATION

PERFORMED ON A VIRTUAL ADDRESS, attorney docket no. RAB 97-001."

IN THE CLAIMS:

Please cancel claims 6 - 37.

Please amend claims 1-5 as follows:

1. (Amended) A circuit for storing physical address translation information to reduce address translation time in a computer system, said circuit comprising:

a) a data path for receiving a virtual address, said virtual address including a segment identifier for identifying a segment [identifier] and a segment offset; and

b) a segment descriptor memory coupled to said data path and selectable by said segment identifier, said memory capable of storing at least the following:

i) linear address information describing the base of the segment,

ii) linear address information describing the limit of the segment, and

iii) a page frame describing at least a portion of a physical address of said segment; and

wherein a tentative memory reference can be initiated based on the virtual address and the information in the segment descriptor memory and without performing a virtual to linear to physical address translation.

2. (Amended) The circuit of claim 1, wherein the information in the segment descriptor memory is stored in one or more registers, and such information can be combined with a portion of the segment offset to create a physical address that is used to initiate the tentative memory reference.

3. (Amended) The circuit of claim 1, wherein the information in the segment descriptor memory is stored in a cache and such information can be combined with a portion of the segment offset to create a physical address that is used to initiate the tentative memory reference.

4. (Amended) The circuit of claim 1, further including a physical address register coupled to the segment descriptor memory for storing a physical address used to initiate the tentative memory reference, said physical address being comprised of the page frame from said segment descriptor memory and a page offset.

5. (Amended) The circuit of claim [4] 1, wherein the [physical address stored in the physical address register is used to perform a memory access] page frame stored in the segment descriptor memory is based on a prior virtual address.

Please add new claims 38 - 81:

38. A system for performing address translations, said system generating an actual physical address from a virtual address in a time period T, by calculating a linear address based on said virtual address, and by calculating said actual physical address based on said calculated linear address, said system further including:

a fast physical address generator for generating a fast physical address related to said virtual address in a time < T.

39. The system of claim 38, wherein the fast physical address can be used for generating a memory access faster than a memory access based on said actual physical address.

40. The system of claim 39, including a cancellation circuit for cancelling the memory access if the fast physical address and actual physical address are different.

41. The system of claim 38, wherein the fast physical address is generated based on a combination of physical address information from a different virtual address, and partial linear address information relating to said virtual address.

42. The circuit of claim 38, wherein the fast physical address is generated before said calculated linear address.

43. A system for performing address translations using a first operation to convert a first virtual address to a first linear address, and a second operation to convert said first linear address to a first physical address, said system further including:

a tentative physical address generator for generating a tentative physical address related to said first virtual address;

wherein the tentative physical address can be generated before said second operation has completed converting said first linear address.

44. The system of claim 43, wherein the tentative physical address can be used for generating a memory access which is faster than a memory access resulting from said first physical address.

45. The system of claim 44, including a cancellation circuit for cancelling the memory access if the tentative physical address and first physical address are different.

46. The system of claim 43, wherein the tentative physical address is generated based on a combination of prior physical address information and partial linear address information relating to said first virtual address.

47. The circuit of claim 43, wherein the tentative physical address is generated before said first operation has completed converting said first virtual address into said first linear address.

48. The circuit of claim 43, wherein said first virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address.

49. A system for performing address translations using a first operation to convert virtual addresses to linear addresses, and a second operation to convert said linear addresses to physical addresses, said system further including:

a fast physical address generator for generating fast physical addresses related to said virtual addresses;

wherein the fast physical addresses can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

50. The system of claim 49, wherein the fast physical addresses can be used for generating memory accesses faster than memory accesses resulting from said calculated physical addresses.

51. The system of claim 50, including a cancellation circuit for cancelling the memory accesses if the fast physical addresses and calculated physical addresses are different.

52. The system of claim 49, wherein the fast physical addresses are generated based on a combination of physical address information and partial linear address information relating to said virtual addresses.

53. The system of claim 49, wherein said virtual addresses are partially converted to linear addresses by the fast physical address circuit and are combined with physical address information relating to prior virtual addresses to generate the tentative physical addresses.

54. A system for performing address translations comprising:

a virtual to linear address converter circuit for generating a calculated linear address based on a virtual address; and

a linear to physical address converter circuit for generating a calculated physical address based on the calculated linear address, the calculated physical address including a calculated page frame and a calculated page offset; and

a fast physical address circuit for generating a fast physical address including a fast page frame and a fast page offset;

wherein a memory reference can be generated based on the fast physical address.

55. The system of claim 54, wherein the fast physical address is based on linear address information relating to the virtual address and physical address information relating to a prior virtual address.

56. The system of claim 54, wherein the virtual address is partially converted to a linear address by the fast physical address circuit and is combined with physical address information relating to a prior virtual address to generate the tentative physical address.

57. A system for performing address translations using a first operation to convert a first virtual address to a first linear address, and a second operation to convert said first linear address to a first physical address, the system further including:

an address translation memory, accessible by said system while said first operation is converting said first virtual address, and capable of storing prior physical address information generated during a prior address translation by said second operation based on a prior virtual address;

wherein a fast physical address can be generated based on the prior physical address information and said first linear address before said second operation has completed converting said first linear address.

58. The system of claim 57, wherein the fast physical address can be used for an accelerated memory access which is faster than a memory access resulting from said first physical address.

59. The system of claim 58, including a cancellation circuit for cancelling the fast memory access if the fast physical address and first physical address are different.

60. The system of claim 57, wherein the fast physical address is comprised of:

(iii) a page frame portion based on the prior physical address information; and

(iv) a page offset portion based on the result of converting said first virtual address to a first linear address.

61. A system for performing address translations comprising:

an address translation memory capable of storing:

(i) a portion of a physical address corresponding to a stored page frame; and

(ii) segment base information relating to a virtual address; and

a virtual to linear address converter circuit for generating a calculated linear address based on combining a portion of the virtual address and the segment base; and

a linear to physical address converter circuit for receiving and generating a calculated physical address based on the calculated linear address, the calculated physical address including a first page frame and a first page offset; and

a fast physical address circuit for generating a fast physical address comprised of the stored page frame combined with a fast page offset portion derived from the segment base and the virtual address;

wherein the fast physical address is calculated prior to the generation of said calculated physical address.

62. The system of claim 61, wherein the fast physical address can be used for generating a fast memory access which is generated more quickly than a memory access resulting from said first physical address

63. The system of claim 61, including a cancellation circuit for cancelling the fast memory access if the fast physical address and first physical address are different.

64. The circuit of claim 61, wherein the fast physical address is generated prior to the generation of the first linear address.

65. The system of claim 61, wherein the stored page frame is generated in a prior address translation based on a prior virtual address.

66. A method of performing a translation of a virtual address in a computer system, said method including the steps of :

(a) calculating a fast physical address related to said virtual address; and

(b) calculating a linear address based on said virtual address; and

(c) calculating an actual physical address based on the linear address;

wherein step (a) is completed prior to the completion of step (c), and the fast physical address can be used to initiate a fast memory reference.

67. The method of claim 66, further including a step (d): cancelling the memory access if the fast physical address and actual physical address are different.

68. The method of claim 66, wherein the fast physical address is generated based on a combination of physical address information from a different virtual address, and partial linear address information relating to said virtual address.

69. The method of claim 66, wherein step (a) is completed prior to the completion of step (b).

70. A method of generating memory references based on virtual addresses in a computer system, said method including the steps of:

(a) generating tentative memory references based on said virtual addresses; and

(b) converting said virtual addresses to linear addresses during a segmentation operation; and

(c) converting said linear addresses to physical addresses during a paging operation, so that actual memory references can be made based on said physical addresses;

wherein the tentative memory reference can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

71. The method of claim 70, further including a step (d): cancelling the tentative memory reference if the tentative memory reference and actual memory reference are different.

72. The method of claim 70, wherein the tentative memory reference is generated based on a combination of physical address information and partial linear address information relating to said virtual addresses.

73. The method of claim 70, wherein step (a) is completed prior to the completion of step (b).

74. A method of generating a fast memory reference using a fast physical address derived from a virtual address in a computer system, the method including the steps of:

(a) converting a portion of said virtual address into a partial linear address; and

(b) combining the partial linear address with physical address information obtained from a prior memory reference to generate said fast physical address;

(c) generating a memory reference based on the fast physical address;

(d) converting said virtual address into an actual physical address;

(e) cancelling the memory reference if the fast physical address and actual physical address are different.

75. The method of claim 74, wherein the fast physical address is generated prior to the generation of the linear address.

76. The method of claim 74, wherein the fast physical address is used to generate a fast memory access prior to the generation of the linear address.

77. A method of generating physical addresses from virtual addresses in a computer system, the method including the steps of:

(a) generating a first calculated linear address based on a first virtual address in a first operation; and

(b) generating a fast physical address in a second operation, the fast physical address including linear address information relating to said first virtual address and portions of physical address information relating to said first virtual address; and

(c) generating a first calculated physical address in a third operation based on the first calculated linear address;

wherein the fast physical address is generated prior to the generation of the first calculated physical address.

78. The method of claim 77, wherein the fast physical address is used to generate a tentative memory access prior to the generation of the first calculated physical address.

79. The method of claim 78, including a step (d): cancelling the tentative memory access if the fast physical address and first calculated physical address are different.

80. The method of claim 79, further including a step (e): generating a memory access request based on the first calculated physical address; and (f) storing physical address information relating to the first calculated physical address for use in a later address translation.

81. The method of claim 77, wherein the first and second operations overlap in time, and the fast physical address is generated prior to the generation of the first calculated linear address.

Remarks

Original claims 6 - 36 have been canceled. Claims 1-5 and 38 - 81 are presently pending.

Original claims 1-5 have been amended to specify that the linear address information and page frame information in the segment descriptor memory can be used to initiate a tentative memory reference, and without performing a virtual-linear-physical address translation. These features are neither disclosed nor suggested in Crawford '836, nor in any of the references submitted or considered to date. Since this feature is neither disclosed nor suggested by any of the art considered to date, applicant submits that the above claims are allowable at this time.

All of the new claims submitted herewith (38 - 81) are distinguishable over the art as well, since applicants believe they are the first to disclose and generally describe a *fast* physical address generator for use in a virtual-linear-physical address translation environment.

Enclosed also is an IDS for all the material references known to applicant. Most of these references were already considered in the parent application. For those references not considered, (1) EP 0-668-565 (Kranich); (2) U.S. Patent No. 4,400,774 (Toy); and (3) U.S. Patent No. 5,165,028, applicant submits the following explanation of why the present invention distinguishes thereover:

First, the Kranich reference does not make any mention of a combination of segmentation and paging based address translation, or a virtual-linear-physical conversion of any kind. In fact, it seems that a different type of address translation is described because at column 9, ll. 55 - column 10, ll. 35 it discusses a virtual address in the form of a virtual "page" system, which is not a "virtual address" of the segment identifier plus segment offset variety disclosed and claimed in the present invention. See present disclosure at p. 2, ll. 4-15 (discussing segmentation). In Kranich the virtual address is disclosed to already contain the page offset (see c.10, ll. 15-18) and therefore no segmentation (virtual to linear address conversion) operation is ever disclosed to occur. Hence there is no "segment," no "segment identifier," no "segmentation" unit or circuit, no "linear address," etc. etc., as those terms are used in the claims. Moreover, as indicated in the attached supplemental IDS, while Kranich claims a U.S. priority date prior to the present application (and may conceivably become 102(e) art at some point), the above European application was not published until after the filing date of the present invention, and therefore it is not believed to be prior art to the present application at this time. Since its status cannot be confirmed at this time, however, applicant requests that it be considered in the present application.

The Toy reference seems deficient for similar reasons. The Zulian reference also makes

mention of a logical (virtual) address that has an offset field that is the <u>same</u> as the physical address, see column 3, ll. 55-59, (like the <u>Kranich</u> virtual paging), and this disclosure is similarly different from the virtual addresses used in the present invention. The <u>Toy</u> reference construction of the virtual address also does not result in a virtual - linear - physical address conversion of the type generally described in the present disclosure.

In short, the present invention solves the timing bottleneck created by repetitive virtual-linear-physical address calculations in segmentation plus optional paging address translation systems (such as shown in Crawford '836) by using a <u>tentative</u> memory reference and this concept is neither disclosed nor suggested in any of the above references. Accordingly, the present claims are distinguishable over these newly submitted references as well, and should be allowable.

Should the Examiner believe it is necessary or fruitful to discuss any of the above points in person, Applicant is open to a teleconference (408-342-1862) at any convenient time.

Respectfully submitted,

Law+

Date: August 4, 1997

J. Nicholas Gross, Attorney, Reg. No. 34,175

*I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Commissioner of Patents and Trademarks, this 4th of August, 1997.*

OIPE JC
MAR 2 2 2000
PATENT & TRADEMARK

**UNITED STATES DEPARTMENT OF COMMERCE**
**Patent and Trademark Office**
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, DC 20231

(Su)

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | | ATTORNEY DOCKET NO. |
|---|---|---|---|---|
| 08/905,356 | 08/04/97 | BELGARD | R | RAB-97-002 |

LM02/1016

PETER COURTURE
LAW+
993 HIGHLAND CIRCLE
LOS ALTOS CA 94024

| EXAMINER | |
|---|---|
| NGUYEN, T | |
| **ART UNIT** | **PAPER NUMBER** |
| 2751 | 6 |

DATE MAILED: 10/16/98

**Please find below and/or attached an Office communication concerning this application or proceeding.**

**Commissioner of Patents and Trademarks**

PTO-90C (REV. 2/95)

2-Mail Copy

| **Office Action Summary** | Application No. 08/905,356 | Applicant(s) Belgard |
|---|---|---|
| | Examiner T Nguyen | Group Art Unit 2751 |

***—The MAILING DATE of this communication appears on the cover sheet beneath the correspondence address—***

**Period for Response**

A SHORTENED STATUTORY PERIOD FOR RESPONSE IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a response be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for response is specified above, such period shall, by default, expire SIX (6) MONTHS from the mailing date of this communication .
- Failure to respond within the set or extended period for response will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).

**Status**

☒ Responsive to communication(s) filed on 8-4-97.

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle,* 1935 C.D. 1 1; 453 O.G. 213.

**Disposition of Claims**

☒ Claim(s) 1-5, 38-81 is/are pending in the application.

Of the above claim(s) ______ is/are withdrawn from consideration.

☐ Claim(s) ______ is/are allowed.

☒ Claim(s) 1-5, 38-81 is/are rejected.

☐ Claim(s) ______ is/are objected to.

☐ Claim(s) ______ are subject to restriction or election requirement.

**Application Papers**

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The proposed drawing correction, filed on ______ is ☐ approved ☐ disapproved.

☐ The drawing(s) filed on ______ is/are objected to by the Examiner.

☒ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. § 119 (a)-(d)**

☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 11 9(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) ______.

☐ received in this national stage application from the International Bureau (PCT Rule 1 7.2(a)).

*Certified copies not received: ______.

**Attachment(s)**

☒ Information Disclosure Statement(s), PTO-1449, Paper No(s). 5

☐ Interview Summary, PTO-413

☒ Notice of References Cited, PTO-892

☐ Notice of Informal Patent Application, PTO-152

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Other ______

**Office Action Summary**

Application/Control Number: 08/905,356

Art Unit: 2751

## DETAILED ACTION

1. This is a response to the amendment and IDS, field 8/4/97.

2. Claims 1-5,38-81 are pending. Claims 6-37 have been canceled.

### *Information Disclosure Statement*

3. The IDS, filed 8/4/97, has been considered.

### *Specification*

4. The abstract of the disclosure is objected to because it does summarize the invention as claimed. Correction is required. See MPEP § 608.01(b).

### *Claim Rejections - 35 U.S.C. § 112*

5. Claims 38,39,41,42 are rejected under 35 U.S.C. 112, first paragraph for undue breadth. In re Hyatt, 708 F.2d 712,>714 - 715,< 218 USPQ 195>, 197< (Fed. Cir. 1983) (A single means claim which covered every conceivable means for achieving the stated purpose was held nonenabling for the scope of the claim because the specification disclosed at most only those means known to the inventor.). The claimed system only has one mean., the fast physical address generator.

### *Double Patenting*

6. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention,"

in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

7. Claims 1-5 provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-5 of copending Application No. 08/905,410. This is a provisional double patenting rejection since the conflicting claims have not in fact been patented.

### *Claim Rejections - 35 U.S.C. § 103*

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1- 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crawford(US 5,321,836).

**As to claim 1:**

Crawford discloses a virtual memory management method and apparatus using segmentation and optional, independent paging mechanism. He discloses a data path for receiving a virtual address, including a segment identifier and offset(Figure 2). The claimed segment descriptor memory for storing the base address, limit of the physical address, and physical address is taught as address

translation unit which stores the base and limit of the address and also the generated physical address(Figures 2 & 3, column 4 lines 50-69). Crawford does not specifically teach initiating a memory reference based on the virtual address and the information in the segment descriptor memory. However, Figure 2 illustrates that the virtual address and partial information in the segment descriptor memory is used to generate the physical address the physical memory through adder 26. One of ordinary skills in the art at the time of the invention would realize that the combination of the virtual address and the partial information in the segment descriptor memory can be used to initiate access the memory. Thus it would have been obvious to one of ordinary skills in the art that a memory reference can be initiated knowing the virtual address and partial information of the segment descriptor memory.

10. **As to claims 2,3:**

Crawford discloses the segment memory being a plurality of registers(Abstract, column 4 line 35+). Figure 2 illustrates that the virtual address and information in the segment descriptor memory is used to generate the physical address the physical memory through adder 26. He also discloses that the descriptor memory could be cache memory(column 5 line 12+).

11. **As to claims 4,5:**

Crawford teaches the physical address register storing a physical address comprising of a page frame and offset(Figure 4). It is also noted that Applicant's Figure 1 also teaches this claim limitation. The resulting physical address is used to access physical memory(column 1 line 24,

column 3 line 35+). He teaches using a 4 bit attribute field as information describing if the page can be used for translation(column 5 lines 28-38).

12. Claims 38-81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crawford(US 5,321,836) in view of Toy(US 4,400,774).

**As to claims 38,39,42-44,47-50,53,54,56-58,60-62,64-66,69,70,73,75-78,81:**

Crawford teaches the address translation system as claimed(see response to claim 1) but does not specifically teach calculating a physical address faster than generating an actual physical address. Toy discloses a nonsegmented memory system with a speculative address generator as cache address unit 125, which uses previous address bits to predict new physical addresses(col 3 ln 54 - col 4 ln 25, Figure 1). This provides for a faster physical address generation if the desired physical address shares the same address bits. The fast physical address is generated before the linear address is calculated(Crawford Figure 2). The speculative address can be used for generating a faster memory access. It would have been obvious to one of ordinary skills in the art at the time of the invention to use Toy's speculative address generator in Crawford's segmented memory system so that physical addresses can be generated more quickly.

**As to claims 40,45,51,59,63,67,71,74,79:**

Toy teaches canceling the memory access if the fast physical address does not equal the actual physical address(col 5 lns 13-49).

**As to claims 41,46,52,55,68,72,80:**

Toy teaches the fast physical address is generated based on combination of physical address information from a different virtual address, and partial linear address information relating to the virtual address by using previous address bits to generate the fast physical address(col 4 lns 1-36).

### *Conclusion*

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Than Nguyen whose telephone number is (703) 305-3866.

14. Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Than Nguyen
October 9, 1998

Eddie Chan

EDDIE P. CHAN
SUPERVISORY PATENT EXAMINER

## *Notice of References Cited*

| Application No. | Applicant(s) |
|---|---|
| 08/905,350 | Belgard |

| Examiner | Group Art Unit | Page |
|---|---|---|
| T Nguyen | 2751 | Page 1 of 1 |

### U.S. PATENT DOCUMENTS

| * | | DOCUMENT NO. | DATE | NAME | CLASS | SUBCLASS |
|---|---|---|---|---|---|---|
| * | A | 4,400,770 | 8/1981 | Ing | 711 | 3 |
| * | B | 5,321,836 | 6/1990 | Crawford | 711 | 206 |
| | C | | | | | |
| | D | | | | | |
| | E | | | | | |
| | F | | | | | |
| | G | | | | | |
| | H | | | | | |
| | I | | | | | |
| | J | | | | | |
| | K | | | | | |
| | L | | | | | |
| | M | | | | | |

OIPE JC69 MAR 22 2000 PATENT & TRADEMARK OFFICE

### FOREIGN PATENT DOCUMENTS

| * | | DOCUMENT NO. | DATE | COUNTRY | NAME | CLASS | SUBCLASS |
|---|---|---|---|---|---|---|---|
| | N | | | | | | |
| | O | | | | | | |
| | P | | | | | | |
| | Q | | | | | | |
| | R | | | | | | |
| | S | | | | | | |
| | T | | | | | | |

### NON-PATENT DOCUMENTS

| * | | DOCUMENT (Including Author, Title, Source, and Pertinent Pages) | DATE |
|---|---|---|---|
| | U | | |
| | V | | |
| | W | | |
| | X | | |

* A copy of this reference is not being funished with this Office action.
(See Manual of Patent Examining Procedure, Section 707.05(a).)

#5/IDS
Approval
3-28-97

SHEET 1 OF 2

| FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE<br>LIST OF PRIOR ART CITED BY APPLICANT (Use several sheets if necessary) | ATTY. DOCKET NO. RAB 97-001 | SERIAL NO. 08/905,356 ~~Not Assigned~~ |
|---|---|---|
| | APPLICANT Richard A. Belgard | |
| | FILING DATE 08/04/97 | GROUP: ~~2309~~ 2757 |

08/905356
08/04/97
68718 U.S. PTO

## U. S. PATENT DOCUMENTS

| EXAMINER INITIAL | DOCUMENT NUMBER | DATE | NAME | CLASS | SUBCLASS | FILING DATE IF APPROPRIATE |
|---|---|---|---|---|---|---|
| NTV | 4 0 8 4 2 2 5 | 4/11/78 | Anderson et. al | ~~364~~ 711 | ~~200~~ 206 | 9/24/76 |
| NTV | 4 4 0 0 7 7 4 | 8/23/83 | Toy | ~~364~~ 711 | ~~200~~ 3 | 2/2/81 |
| NTV | 5 1 6 5 0 2 8 | 11/17/92 | Zulian | ~~395~~ 711 | ~~400~~ 3 | 3/7/89 |
| NTV | 5 3 2 1 8 3 6 | 6/14/94 | Crawford, et al. | ~~395~~ 711 | ~~400~~ 206 | 4/9/90 |
| NTV | 5 3 3 5 3 3 3 | 8/2/94 | Hinton, et al. | ~~395~~ 711 | ~~400~~ 207 | 10/29/91 |
| NTV | 5 4 2 3 0 1 4 | 6/6/95 | Hinton, et al. | ~~395~~ 711 | ~~400~~ 3 | 2/24/94 |

## FOREIGN PATENT DOCUMENTS

| | DOCUMENT NUMBER | DATE | COUNTRY | CLASS | SUBCLASS | TRANSLATION YES | TRANSLATION NO |
|---|---|---|---|---|---|---|---|
| NTV | 0 6 6 8 5 6 5 | 8/23/95 | EPO | G06F | 12/10 | NONE | |

## OTHER PRIOR ART (Including Author, Title, Date, Pertinent Pages, Etc.)

| | | | |
|---|---|---|---|
| | | | *Intel Microprocessors*, Vol. 1, Intel Corporation, 1993, pp. 2-229 to 2-287 |
| NTV | | | *Computer Architecture A Quantitative Approach*, Hennessey and Patterson, pp. 432-497 |
| NTV | | | *U5S 486 Green CPU*, United Microelectronics Corporation, 1994-95, pp. 3-1 to 3-26 |

| EXAMINER | DATE CONSIDERED |
|---|---|
| TM | 10/8/98 |

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

| FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE | ATTY. DOCKET NO. RAB 97-001 | SERIAL NO. 08/905,356 ~~Not Assigned~~ |
|---|---|---|
| LIST OF PRIOR ART CITED BY APPLICANT *(Use several sheets if necessary)* | APPLICANT **Richard A. Belgard** | |
| | FILING DATE **08/04/97** | GROUP: ~~2309~~ 2751 |

## U. S. PATENT DOCUMENTS

| EXAMINER INITIAL | DOCUMENT NUMBER | DATE | NAME | CLASS | SUBCLASS | FILING DATE IF APPROPRIATE |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

## FOREIGN PATENT DOCUMENTS

| | DOCUMENT NUMBER | DATE | COUNTRY | CLASS | SUBCLASS | TRANSLATION YES | TRANSLATION NO |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

## OTHER PRIOR ART *(Including Author, Title, Date, Pertinent Pages, Etc.)*

| | | | |
|---|---|---|---|
| NTV | | | *The Multics System*, Elliot Organick, 1972, pp. 6-7, 38-51 |
| NTV | | | *DPS-8 Assembly Instructions*, Honeywell Corporation, April, 1980, Chapters 3 and 5 |
| | | | |
| | | | |

| EXAMINER | DATE CONSIDERED |
|---|---|
| TNguyen | 10/8/98 |

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

US005321836A

# United States Patent [19]

Crawford et al.

[11] Patent Number: 5,321,836

[45] Date of Patent: Jun. 14, 1994

[54] VIRTUAL MEMORY MANAGEMENT METHOD AND APPARATUS UTILIZING SEPARATE AND INDEPENDENT SEGMENTATION AND PAGING MECHANISM

[75] Inventors: John H. Crawford, Santa Clara; Paul S. Ries, San Jose, both of Calif.

[73] Assignee: Intel Corporation, Santa Clara, Calif.

[21] Appl. No.: 506,211

[22] Filed: Apr. 9, 1990

Related U.S. Application Data

[63] Continuation of Ser. No. 185,325, Apr. 19, 1988, Pat. No. 4,972,338, which is a continuation of Ser. No. 744,389, Jun. 13, 1985, abandoned.

[51] Int. Cl.$^5$ .............................................. G06F 12/10

[52] U.S. Cl. .................................... 395/400; 395/425; 364/DIG. 1; 364/243; 364/243.4; 364/243.41; 364/256.3; 364/256.4; 364/256.5

[58] Field of Search ..................... 365/238.5; 395/400, 395/425, 800

[56] References Cited

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 3,047,228 | 7/1962 | Bauer et al. | 364/700 |
| 3,292,152 | 12/1966 | Barton | 365/49 |
| 3,298,001 | 1/1967 | Couleur et al. | 395/425 |
| 3,303,477 | 2/1967 | Voight | 395/400 |
| 3,351,913 | 11/1967 | Pine | 395/425 |
| 3,370,274 | 2/1968 | Kettley et al. | 395/775 |
| 3,412,382 | 11/1968 | Couleur et al. | 395/400 |
| 3,425,039 | 1/1969 | Bahrs et al. | 395/425 |
| 3,437,998 | 4/1969 | Bennett et al. | 395/425 |
| 3,454,931 | 7/1969 | Bahrs et al. | 395/400 |
| 3,461,434 | 8/1969 | Barton et al. | 395/400 |
| 3,473,159 | 10/1969 | Cantrell et al. | 395/425 |
| 3,479,649 | 11/1969 | Bahrs et al. | 395/725 |
| 3,510,847 | 5/1970 | Carlson et al. | 395/400 |
| 3,521,240 | 7/1970 | Bahrs et al. | 395/425 |
| 3,525,080 | 8/1970 | Couleur et al. | 395/425 |
| 3,546,677 | 12/1970 | Barton et al. | 395/400 |
| 3,548,384 | 12/1970 | Barton et al. | 395/400 |
| 3,601,809 | 8/1971 | Gray et al. | 395/425 |
| 3,649,767 | 3/1972 | Muroga et al. | 379/279 |
| 3,748,653 | 7/1973 | Debruyne et al. | 365/200 |
| 3,761,902 | 9/1973 | Weinberger | 365/49 |
| 3,764,996 | 10/1973 | Ross | 395/425 |
| 3,786,427 | 1/1974 | Schmidt et al. | 395/425 |
| 3,787,813 | 1/1974 | Cole et al. | 395/400 |
| 3,818,460 | 6/1974 | Beard et al. | 395/400 |
| 3,839,706 | 10/1974 | Borchsenius | 395/400 |
| 3,891,974 | 6/1975 | Coulter et al. | 395/500 |
| 3,916,385 | 10/1975 | Parmar et al. | 395/425 |
| 3,938,096 | 2/1976 | Brown et al. | 395/400 |
| 3,950,730 | 4/1976 | Bienvenu et al. | 395/425 |
| 3,953,717 | 4/1976 | Rottier et al. | 371/17 |
| 4,060,849 | 11/1977 | Bienvenu et al. | 395/275 |
| 4,068,300 | 1/1978 | Bachman | 395/600 |
| 4,077,058 | 2/1978 | Appell et al. | 395/650 |
| 4,079,453 | 3/1978 | Dahl | 395/400 |
| 4,092,715 | 5/1978 | Scriver | 395/400 |
| 4,128,875 | 12/1978 | Thurber et al. | 395/425 |
| 4,145,734 | 3/1979 | Bienvenu et al. | 395/575 |
| 4,173,783 | 11/1979 | Couleur et al. | 395/275 |
| 4,177,510 | 12/1979 | Appell et al. | 395/425 |
| 4,279,014 | 7/1981 | Cassonnet et al. | 395/425 |
| 4,300,192 | 11/1981 | Couleur et al. | 395/425 |

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

1209709 8/1986 Canada .

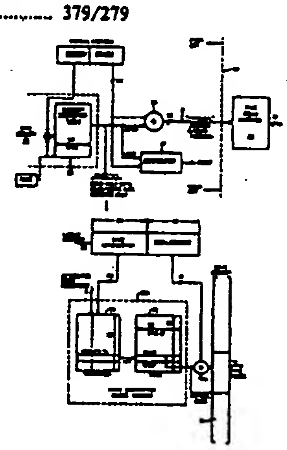
OTHER PUBLICATIONS

A. Brandwajn et al., "A Study of a Page-on-Demand System", Information Processing Letters, vol. 6, No. 4, Aug. 1977, pp. 125-132.

[57] ABSTRACT

Microprocessor architecture for an address translation unit which provides two levels of memory management is described. Segmentation registers and an associated segmentation table in main memory provide a first level of memory management which includes attribute bits used for protection, priority, etc. A page cache memory and an associated page directory and page table in main memory provide a second level of management with independent protection on a page level.

6 Claims, 8 Drawing Sheets



# United States Patent [19]

**Toy**

[11] **4,400,774**

[45] **Aug. 23, 1983**


[54] **CACHE ADDRESSING ARRANGEMENT IN A COMPUTER SYSTEM**


[75] Inventor: **Wing N. Toy**, Glen Ellyn, Ill.

[73] Assignee: **Bell Telephone Laboratories, Incorporated**, Murray Hill, N.J.



[21] Appl. No.: **230,893**

[22] Filed: **Feb. 2, 1981**

[51] **Int. Cl.**³ .............................................. **G06F 9/06**

[52] **U.S. Cl.** .............................................. **364/200**

[58] **Field of Search** ... 364/200 MS File, 900 MS File


[56] **References Cited**

U.S. PATENT DOCUMENTS

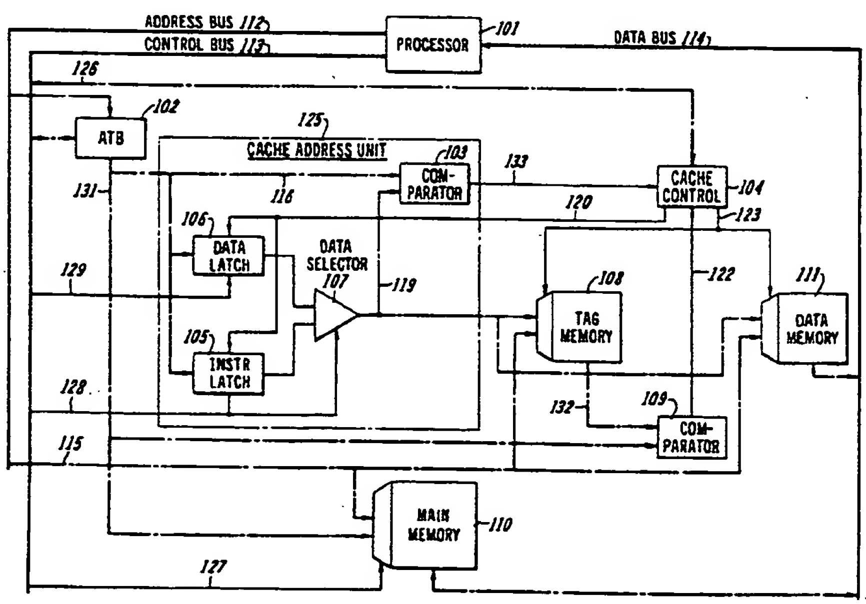

| | | | |
|---|---|---|---|
| 3,693,165 | 9/1972 | Reiley et al. | 364/200 |
| 3,723,976 | 3/1973 | Alvarez et al. | 364/200 |
| 3,761,881 | 9/1973 | Anderson et al. | 364/200 |
| 3,786,427 | 1/1974 | Schmidt et al. | 364/200 |
| 3,825,904 | 7/1974 | Burk et al. | 364/200 |
| 3,979,726 | 9/1976 | Lange et al. | 364/200 |
| 4,084,230 | 4/1978 | Matick et al. | 364/200 |
| 4,096,573 | 6/1978 | Heller et al. | 364/200 |
| 4,128,875 | 12/1978 | Thurber et al. | 364/200 |


*Primary Examiner*—Harvey E. Springborn
*Attorney, Agent, or Firm*—P. Visserman

[57] **ABSTRACT**


In a computer system having a cache memory and using virtual addressing, effectiveness of the cache is improved by storing a subset of the least significant real address bits obtained by translation of a previous virtual address and by using this subset in subsequent cache addressing operations. The system functions in the following manner. In order to access a memory location in either the main memory or cache memory, a processor generates and transmits virtual address bits to the memories. The virtual address bits comprise segment, page and word address bits. The word address bits do not have to be translated, but an address translation buffer (ATB) translates the segment and page address into real address bits. A subset of the least significant bits of the latter word address bits represent the address needed for accessing the cache. In order to increase cache memory performance, the cache memory comprises a cache address unit which stores the subset of the real address bits from the ATB. These stored address bits are used in subsequent operations along with the word address bits for accessing the cache memory until the stored address bits no longer equal the current subset of least significant real address bits transmitted from the ATB. When the stored address bits no longer equal the current subset, the cache address unit then stores the current subset; and the cache memory is reaccessed utilizing the word address bits and current subset.


**7 Claims, 1 Drawing Figure**



RAB 97-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: ) Art Unit: *2751*
*Belgard, R.* )
) Examiner: *Nguyen, T*
Serial No.: *08/905,356* )
)
Filed: *8/4/97 as continuation of parent application* )
*serial no. 08/458,479 filed 6/2/95* )
)
For: *Fast Address Generator For Reducing* )
*Virtual-Linear-Physical Address Conversion Time* )
------------------------------------------------------- )

AMENDMENT AND
RESPONSE TO OFFICE ACTION

Honorable Commissioner
of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

Applicant respectfully submits the following amendment and response to place this case in condition for allowance:

IN THE SPECIFICATION:

**Please amend the Abstract as follows:**

An improved address translation method and mechanism for memory management in a computer system is disclosed. [A segmentation mechanism employing segment registers maps virtual addresses into a linear address space. A paging mechanism optionally maps linear addresses into physical or real addresses. Independent protection of address spaces is provided at each level. Information about the state of real memory pages is kept in segment registers or a segment register cache potentially enabling real memory access to occur simultaneously with address calculation, thereby increasing performance of the computer system.] <u>A fast physical address is generated in parallel with a fully computed virtual-linear-physical address in a system using segmentation and optional paging. This fast physical address is used for a tentative or speculative memory reference, which reference can be canceled in the event the fast physical address does not match the fully computed address counterpart. In this manner, memory references can be accelerated in a computer system by avoiding a conventional translation scheme requiring two separate and sequential address translation operations – i.e. from virtual to linear, and from linear to physical</u>.

<u>IN THE CLAIMS</u>

**Please cancel claims 1 – 5.**

**Please amend claim 38:**

38. A system for performing address translations, [said system generating an actual physical address from a virtual address in a time period T, by calculating a linear address based on said virtual address, and by calculating said actual physical address based on said calculated linear address, said system further including] <u>comprising</u>:

<u>means for generating an actual physical address from a virtual address in a time period T, by calculating a linear address based on said virtual address, and by calculating said actual physical address based on said calculated linear address; and</u>

a fast physical address generator for generating a fast physical address related to said virtual address in a time < T.

**Please add the following claims (82 – 112):**

82. A system for performing memory references in a processor which employs both segmentation and optional paging during an address translation, said system comprising:

    means for performing an address translation by generating a first physical address from a first virtual address by first calculating a first linear address based on a first segment identifier and first offset associated with the first virtual address, and then calculating the first physical address based on the first calculated linear address; and

    a fast physical memory access circuit for generating a fast memory reference, which fast memory reference is based on physical address information from said address translation means;

    a bus interface circuit for initiating a fast memory access to a memory subsystem based on said fast memory reference.

83. The system of claim 82, further including a comparator for determining whether said fast memory reference can be used for a fast memory access.

84. The system of claim 83, further including a cancellation circuit for canceling said fast memory access.

85. The system of claim 84, wherein the system performs an actual memory reference after said fast memory reference is cancelled.

86. A method for performing memory accesses between a processor and a memory, said processor having virtual addresses that are segmented and optionally paged, the method comprising the steps of:

generating computed physical addresses by converting virtual addresses having a segment identifier and a segment offset into linear addresses, and then converting said linear addresses into a physical addresses;

generating a speculative physical address based on one of said computed physical addresses;

initiating a speculative memory access based on said speculative physical address.

87. The method of claim 86, further including a step of initiating an actual memory access based on a physical address which has been computed during separate segmentation and paging operations.

88. The method of claim 87, wherein said speculative memory access is completed unless canceled in favor of an actual memory access.

89. A system for performing a first and a second address translation of first and second virtual addresses respectively, the system comprising:

a virtual to linear address converter circuit generating a first calculated linear address based on the first virtual address; and

a linear to physical address converter circuit completing the first address translation by generating a first calculated physical address from said first calculated linear address, said first calculated physical address including a first calculated page frame and a first calculated page offset; and

wherein the system uses information from the first address translation during the second address translation so that the second address translation can be performed in less time than the first address translation.

90. The system of claim 89, further including a comparator for determining whether said second address translation can be used for a memory access.

91. The system of claim 89, wherein said second address translation is based on a combination of partial linear address information relating to said second virtual address and physical address information from a different virtual address.

92. The system of claim 89, wherein the system also calculates an actual second physical address from said second virtual address, by calculating a second linear address based on a second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.

93. The system of claim 92, wherein at least a portion of said actual second physical address is compared with a corresponding portion of said second physical address from said fast physical address generator, and when said portions are not equal, said actual second physical address is used for a memory access.

94. The system of claim 89, further including a register for storing address information pertaining to the first virtual address for use during said translation of said second virtual address.

95. A circuit for performing fast translations of virtual addresses to physical addresses in a computer system, the circuit including:

an address generator performing a first address translation of a first virtual address having an associated first segment identifier and a first offset;

said address generator also performing a fast address translation of a second virtual address having an associated second segment identifier and a second offset;

wherein said address generator uses information from the first address translation during the fast address translation so that said translation of said second virtual address takes less time than said first address translation.

96. The system of claim 95, further including a comparator for determining whether said fast address translation can be used for a memory access.

97. The system of claim 95, wherein said fast address translation is achieved based on a combination of partial linear address information relating to said second virtual address and physical address information from said first virtual address.

98. The system of claim 95, wherein the address generator also performs a calculated translation to calculate an actual second physical address from said second virtual address, by calculating a second linear address based on said second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.

99. The system of claim 98, wherein at least a portion of said actual second physical address is compared with a corresponding portion of said second physical address from said fast physical address generator, and when such portions are not equal, said actual second physical address is used for a memory access.

100. The system of claim 95, further including a register for storing address information pertaining to the first virtual address for use during said translation of said second virtual address.

101. A method of translating virtual addresses in a computer system, the method including the steps of:

(a) generating a first calculated physical address based on a first virtual address in a first operation, said first virtual address including a first segment identifier and a first offset; and

(b) generating a second fast physical address in a second operation based on a second virtual address, said second virtual address including a second segment identifier and a second offset, and said second fast physical address being generated based on information obtained during said first operation;

wherein said second operation takes less time to perform in said computer system than said first operation.

102. The method of claim 101, further including a step of determining whether a memory access can be made using said second fast physical address.

103. The method of claim 101, wherein during step (b) said second physical address is generated based on a combination of partial linear address information relating to said second virtual address and physical address information from said first virtual address.

104. The method of claim 101, further including a step (c): generating an actual second physical address from said second virtual address during a third operation, by calculating a second linear address based on said second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.

105. The method system of claim 104, further including step (d): comparing at least a portion of said actual second physical address with a corresponding portion of said second physical address from said fast physical address generator, and when such portions are not equal, using said actual second physical address for a memory access.

106. The system of claim 101, further including a step of storing address information pertaining to the first virtual address in a register during said first operation for use during said second operation.

107. A method of performing address translations in a computer system, the method including the steps of:

(a) performing a first address translation by translating a first virtual address into a first physical address by: (i) first calculating a first linear address based on a first segment identifier and first offset associated with said first virtual address; and (ii) calculating said first physical address based on said first calculated linear address and

(b) performing a second address translation using information obtained during said first address translation to translate a second virtual address into a second physical address; wherein said second translation can be achieved in less time than said first translation.

108. The method of claim 107, further including a step of determining whether a memory access can be made using said second physical address.

109. The method of claim 107, wherein during step (b) said second physical address is generated based on a combination of partial linear address information relating to said second virtual address and physical address information from said first virtual address.

110. The method of claim 107, further including a step (c): generating an actual second physical address from said second virtual address, by calculating a second linear address based on said second segment identifier and second offset associated with said second virtual address, and calculating said second physical address based on said second calculated linear address.

111. The method system of claim 110, further including step (d): comparing at least a portion of said actual second physical address with a corresponding portion of said second physical address from said fast physical address generator, and when such portions are not equal, using said actual second physical address for a memory access.

112. The system of claim 107, further including a step of storing address information pertaining to the first virtual address in a register for use during said second address translation.

Remarks

Claims 38 – 112 are pending. Applicant now responds to the Examiner's objections and rejections as set out paragraph by paragraph in the Office Action.

¶4: The Abstract has been amended to better summarize the invention. Thus, applicant believes that the objection to the specification has been overcome at this time.

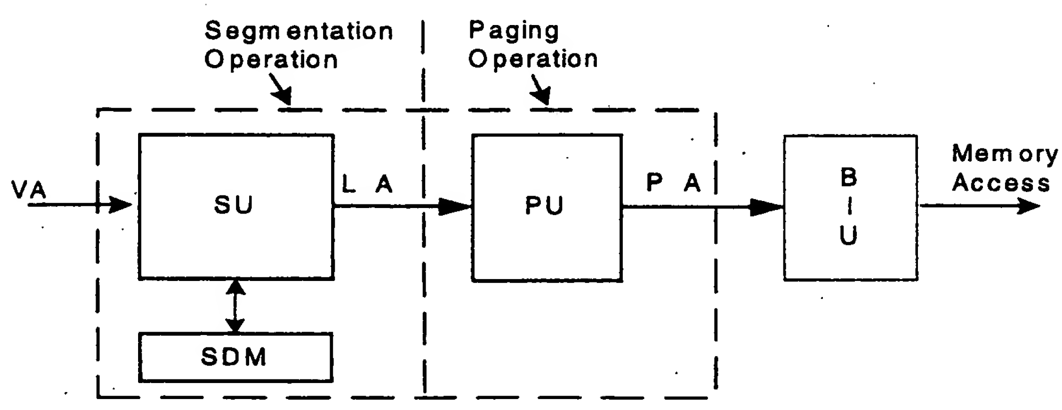
¶5: Claim 38 has been amended. As presently structured, this claim is believed to overcome the Examiner's rejection under § 112 for overbreadth. Claims 39, 41, and 42 depend from claim 38, and thus the present amendment should remove the objections for these claims as well.

¶6 – 7 : Claims 1 – 5 have been canceled without waiver or prejudice. This should obviate the double patenting rejection under § 101.

The following comments are provided in response to the Examiner's rejections of the claims under § 103 in ¶ 8 – 12 of the Office Action.

THE BASIC STRUCTURAL AND OPERATIONAL FEATURES
OF THE PRIOR ART AND THE APPLICANT'S INVENTION

To facilitate the present discussion, applicants have generated the following simplified block diagram of the Crawford '836 reference relied upon by the Examiner, as well as a corresponding simplified block diagram of the present invention:

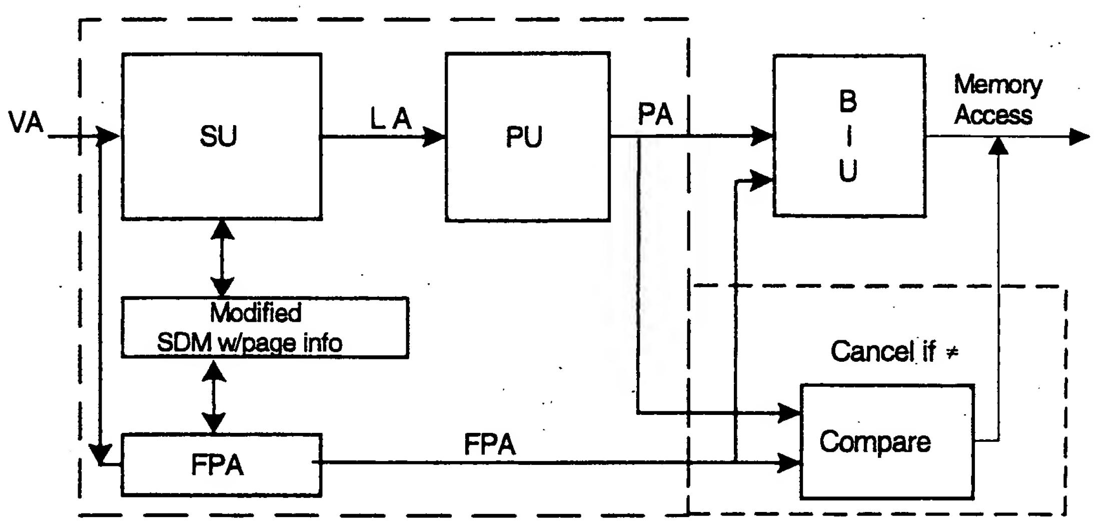


Crawford ATU

The Examiner acknowledges that Crawford '836 does not disclose anything which generates either a "fast" physical address, or a "tentative" or "speculative" memory reference. This is a key and important distinction, because the terms "fast," "tentative" and "speculative" have a plain meaning which is entirely incompatible with the type of structure used by Crawford '836 to generate

physical addresses. Moreover, the Toy reference would not teach one of skill in the art to modify Crawford '836 in any fashion, because the latter specifically teaches away from such modifications as explained below.

The operation of the Crawford reference is demonstrable from the above diagram. Looking at the Crawford '836 Address Translation Unit (ATU) approach, each Virtual Address (VA) is converted by a segmentation unit (SU) during a first operation into a Linear Address (LA) using information from a segment descriptor memory (SDM). The LA is then in turn converted into a Physical address (PA) by a paging unit (PU) in a second operation and then used by a bus interface unit (BIU) to initiate a memory access. This set of operations takes place in the same slow, sequential manner for *every* virtual address, with no regard given to any characteristics of such address, prior address information, etc. It is always a two step operation, and, for each translation, takes the same exact time to complete; therefore it is inefficient in most cases.




## Applicant's ATU

Looking at the applicants' claimed invention, in an Address Translation Unit (ATU) approach, after a first virtual address is converted, the physical address information from such conversion is kept in a modified segment descriptor memory, accessed by segment identifier, which supplies this information to a fast physical address generator. Thereafter, each subsequent Virtual Address (VA) whose segment identifier is the same is converted by a fast physical address generator to create a tentative or speculative address used by a bus interface unit (BIU) to initiate a memory access. In other words, this memory access is initiated extremely rapidly, without waiting for the two-step translation noted above to complete. The *only* time the two-step calculated physical

address is used to access memory is when a portion of such calculated physical address is different from the corresponding portion of the tentative or speculative address. When the two portions are different, the tentative memory access is discontinued by the comparator, and a memory access based on the fully calculated physical address is instead performed.

From the above-simplified pictures and description, is readily apparent how the applicant's claims distinguish over Crawford and the prior art. The modified segment descriptor memory claimed herein, therefore, is quite important, and serves to provide the present invention with functionality (rapid address translation) unavailable in the Crawford type approach. Crawford does not have anything remotely resembling the structure used in the applicant's claimed inventions for storing physical address information in a segment descriptor memory. Nor does it generate tentative or speculative addresses or memory references. There is no mechanism or suggestion for computing a physical address faster from one virtual address to the next. There is no teaching or suggestion for having both fast physical address and a regular address computed at the same time, or in parallel. None of these teachings exist because the Crawford type approach, with strict separation of segmentation and paging, cannot achieve such functionality.

GENERAL DISCUSSION OF PRESENT REJECTIONS

The Examiner nonetheless asserts in ¶ 12 of the Office Action that it is "obvious" to modify Crawford '836 based on the Toy reference to arrive at the applicant's claimed inventions. As the Examiner is well aware nonetheless, for this rejection to be sustainable, there must be some suggestion, teaching, or motivation in Crawford '836, Toy, or some other prior art reference, to perform this modification. And yet, the Examiner *acknowledges* that such information is not to be found in Crawford '836, Toy, or anyplace else for that matter. Crawford nowhere mentions anything except the standard sequential logical-linear-physical calculation for a memory reference. Toy contains no mention or suggestion that the scheme shown therein (which the Examiner acknowledges is "non segmented"[1]) can be applied to a segmented operating system, and more importantly, does not explain how this would be done. Non-segmented address systems are addressed at length in the background of the present disclosure; they are not relevant to the present invention, in large part because they are not subject to the same constraints as a segment plus optional paging type address system as shown in Crawford. It is not surprising, therefore, that Toy

[1] While the Toy reference mentions a "segment" as part of the virtual address, this is clearly a different kind of segment than that discussed by Crawford. This is apparent from the fact that there is no segment ID or segment offset in the virtual address, and by the fact that the virtual address in Toy is never converted to a linear address. Instead the virtual address, which contains paging information, is merely converted in one step to a physical address, which is unlike the virtual-linear-physical type approach disclosed in Crawford.

contains no suggestion for implementing the scheme shown there within an operating system that utilizes separate segment + paging operations.

On this basis alone Applicants submit that the present rejections are improper under applicable case law interpreting § 103, and must be withdrawn. *See e.g, In Re Bond, 15 U.S.P.Q. 2d 1566, 1568-69* (Fed. Cir. 1990) ("obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching, suggestion or incentive supporting the combination."); *In Re Gordon, 221 U.S.P.Q. 1125 (Fed. Cir. 1984)* ("The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification")

The cases are quite clear that the Examiner cannot simply make a bald, conclusory, unsupported statement about what one of skill in the art would find "obvious." *See e.g., In Re Fine, 5 U.S.P.Q. 2d 1596, 1599* (Fed. Cir. 1988) (Board's "bald assertion" that something was within the level of skill in the art is not sufficient evidence to support an obviousness finding). Yet in this case, there is no suggestion cited (and none exists) to support any argument that it might somehow be obvious to combine Crawford '836 with Toy to arrive at the claimed invention. This unsupported analysis suffers other problems: first, it relies on hindsight reconstruction; and second, the Toy reference ATU is *incompatible* with the type of ATU shown in Crawford. In system such as Toy, paging information, including physical address information, is contained within the virtual address. See e.g., the Toy reference at column 3:, ll. 57+ "The virtual address is composed of segment, page, and word address bits." The plain teachings of Crawford show that a virtual address (as well as any segment descriptor tables in a segmentation and optional paging environment) can contain no paging information, or physical address information. This would, in essence, combine segmentation and paging together, and the Crawford '836 reference itself explicitly teaches away from such combination. These facts, too, refute any insinuation that the claimed invention is obvious in light of the art.

## IN REJECTING CLAIMS 37 - 81 THE EXAMINER IMPROPERLY MODIFIES CRAWFORD '836 IN A MANNER INCONSISTENT WITH THE TEACHINGS OF THAT REFERENCE

In ¶ 12 of the Office Action the Examiner relies on an argument that the ATU 20 in Crawford can be modified to include a speculative address generator as described in Toy. Yet, on its face, Crawford '836 is incompatible with the type of address translation logic shown in Toy, which, as mentioned above, uses paging information in the virtual address. Crawford cannot

handle virtual addresses having paging information, and cannot handle paging information during a segmentation operation. *See e.g.*, FIG. 2, the dotted line indicating the demarcation between a segmentation and paging unit. A quick review of the '836 claims confirms that paging information cannot be used during the first part of the address translation (from virtual to linear):

> "...said segment descriptor table describing segments *without reference to whether a segment is paged or unpaged.*" See claim 3, column 10, ll. 9 – 14 (emphasis added).

The other '836 claims have similar limitations requiring that there be *no* paging information in the segmentation descriptor memory, or in the segmentation unit during an address translation. This separation is also categorically enforced in the teachings of the specification, which notes:

> "Segments are defined by a set of segment descriptor tables that are *separate from the page tables* used to describe the page translation." *See* column 3, ll. 11 – 13 (emphasis added).

Thus, Crawford '836 explicitly discourages, and teaches away from, the use of any system, such as that shown in Toy, where segment and paging information is combined.

Applicant was the first to conceive and develop a viable mechanism that provides fast memory references in a segmentation plus optional paging system. There is simply no teaching, suggestion or hint in Toy (or elsewhere) about how to achieve such functionality in such environment. Accordingly, under these circumstances, there is compelling evidence that the claims are non-obvious, and should be allowed.

When a reference explicitly teaches away from the kind of modification proposed by the Examiner, a rejection for obviousness is not proper. *See In Re Fine, supra*, at 1599: "…instead of suggesting that the system be used to detect nitrogen compounds, Eads deliberately seeks to avoid them; it warns against rather than teaches Fine's invention." Similarly, in *In Re Gordon, supra*, 221 U.S.P.Q. at 112, the same conclusion was reached: "…if the French apparatus were turned upside down, it would be rendered inoperable for its intended purpose….[I]n effect, French teaches away from the board's proposed modification." It is crystal clear, from a plain reading of the Crawford reference, that it specifically and actively discourages the use of an address translation system such as shown in Toy, which includes paging information in the virtual address. Crawford is entirely incompatible with such approach, and does not indicate to one of skill in the art how to overcome such incompatibility. Using the virtual address system of Toy, including an accelerated address generated by using paging information, would in fact defeat the whole purpose of the very carefully set up separate segmentation/paging operations described in Crawford. Thus, the present rejections for obviousness should be withdrawn.

There are other important limitations in the claims which the Examiner has also not addressed, or explained why they are obvious in light of the prior art, even if one of skill in the art were to modify Crawford to include the teachings of Toy. The latter, for example, mentions nothing about "linear" addresses, and yet such types of addresses are described in all of the pending claims. Many of the present claims mention the relative timing between the operational steps needed to compute an actual physical address and the operations used to create speculative or tentative memory addresses. Yet, the Toy reference teaches and suggests nothing about this aspect of the invention whatsoever. In large part this is because Toy, as indicated above, is not a virtual-linear-physical address translation system, and therefore does not (and cannot) teach anything to one of skill in the art about the desired timing relationship of such operations. Accordingly, this aspect of many of the claims is not obvious from any of the prior art. This includes claim 43 (which indicates generally that the tentative physical address is generated before the actual linear address is computed), claim 49 (which recites that fast physical addresses can be generated while virtual addresses are being converted into linear addresses), claim 57 (which indicates that a fast physical address for a second can be generated based on prior physical address information and the first linear address before computations for the first linear address have actually been completed); claim 70 (the tentative memory reference can be generated while virtual addresses are being converted into linear addresses. Many of the independent claims include similar limitations: *see, e.g.* claims 44 – 48; 50 – 53; 58 – 60; 64; 69; 70 – 73; 75; 81. This is yet another reason why the claims are distinguishable over the art.

## NEWLY SUBMITTED CLAIMS 38 - 49 ARE PATENTABLE OVER CRAWFORD AS WELL

Newly submitted claims 81 - 111 are also believed to be patentable at this time for the reasons set forth above. For new claims 82 – 85, 89 – 94, and 107 – 112, as discussed above, Toy does not include any suggestion to modify the type of structure shown in Crawford to include a fast memory reference, and does not mention anything about "linear" addresses. This same argument applies with equal force to claims 86 – 88, and 95 – 100, 101 – 106; it is apparent, also, that Toy says nothing about a "segment identifier" or a "segment offset" as called for in these claims.

## CONCLUSION

Consequently, the Examiner's present rejections require that one of skill in the art disregard the plain teachings of the Crawford '836 approach, and then radically modify the ATU shown there with the teachings of Toy to arrive at the claimed invention. This is not an acceptable basis for rejecting the claims under §103, especially since there is no suggestion in either Crawford or Toy to

do so. As the Examiner may not have understood that this type of radical re-engineering of Crawford '836 might be necessary, applicants submit that the present rejections are not supportable.

The present invention therefore is structurally and operationally different from the standard address translation logic shown in Crawford '836. These differences result in a noticeable improvement over Crawford '836, by avoiding repetitive virtual-linear-physical address calculations in segmentation plus optional paging address translation systems. Accordingly, claims are distinguishable over the prior art, including Crawford '836 and Toy, and should be allowable.

A check in the amount of $ 513 has been enclosed to cover the costs of the newly filed claims (6 new independent claims ($39 each) and 31 new claims total ($9 each)).

Should the Examiner believe it that it would be helpful to discuss any of the above points in person, Applicant is open to a telephone conference (408-342-1862) at any convenient time.

These points and others are now discussed in detail.

Respectfully submitted,

Law

Date: November 24, 1998

J. Nicholas Gross, Attorney, Reg. No. 34,175

*I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Commissioner of Patents and Trademarks, this 24th day of November 1998.*

OIPE JCGS
MAR 2 2 2000
PATENT & TRADEMARK OFFICE

FEB 1 8 REC'D

**UNITED STATES DEPARTMENT OF COMMERCE**
**Patent and Trademark Office**
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | | ATTORNEY DOCKET NO. |
|---|---|---|---|---|
| 08/905,356 | 08/04/97 | BELGARD | R | RAD-97-002 |

LM02/0201

PETER COURTURE
LAW+
993 HIGHLAND CIRCLE
LOS ALTOS CA 94024

| EXAMINER | |
|---|---|
| NGUYEN, T | |
| **ART UNIT** | **PAPER NUMBER** |
| 2751 | |

**DATE MAILED:** 02/01/99

**Date Due:** 5.1.99

**Please find below and/or attached an Office communication concerning this application or proceeding.**

**Commissioner of Patents and Trademarks**

PTO-90C (Rev. 2/95)
*U.S. GPO: 1997-417-381/62715

2 - Mail Copy

# Office Action Summary

| Application No. | Applicant(s) |
|---|---|
| 08/905,356 | Belgard R. |
| **Examiner** | **Group Art Unit** |
| T Nguyen | 2751 |

***—The MAILING DATE of this communication appears on the cover sheet beneath the correspondence address—***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, such period shall, by default, expire SIX (6) MONTHS from the mailing date of this communication .
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).

**Status**

☒ Responsive to communication(s) filed on 11-27-98.

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 1 1; 453 O.G. 213.

**Disposition of Claims**

☒ Claim(s) 38-112 is/are pending in the application.

Of the above claim(s) ______ is/are withdrawn from consideration.

☐ Claim(s) ______ is/are allowed.

☒ Claim(s) 38-112 is/are rejected.

☐ Claim(s) ______ is/are objected to.

☐ Claim(s) ______ are subject to restriction or election requirement.

**Application Papers**

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The proposed drawing correction, filed on ______ is ☐ approved ☐ disapproved.

☐ The drawing(s) filed on ______ is/are objected to by the Examiner.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. § 119 (a)-(d)**

☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 11 9(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) ______.

☐ received in this national stage application from the International Bureau (PCT Rule 1 7.2(a)).

*Certified copies not received: ______.

**Attachment(s)**

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). ______

☒ Notice of Reference(s) Cited, PTO-892

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Interview Summary, PTO-413

☐ Notice of Informal Patent Application, PTO-152

☐ Other ______

**Office Action Summary**

Application/Control Number: 08/905,356

Art Unit: 2751

## DETAILED ACTION

### *Response to Amendment*

1. This is a response to the amendment, filed 11/27/98.

2. Claims 38-112 are pending. Claims 82-112 has been added. Claims 1-5 have been canceled.

3. The submitted abstract is acceptable.

4. The rejections to claims 38-42 under 35 U.S.C. 112, first paragraph, as withdrawn in view of Applicant's amendment to overcome the 112 rejection.

### *Response to Arguments*

5. Applicant's arguments with respect to claims 38-112 have been considered but are moot in view of the new ground(s) of rejection. After careful examination of the claims and their scope, the Examiner has made new art rejections to the claims due to their large breadth of scope.

### *Claim Rejections - 35 U.S.C. § 102*

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371© of this title before the invention thereof by the applicant for patent.

7. Claims 38-112 are rejected under 35 U.S.C. 102(e) as being anticipated by Toy(US 4,400,774).

**As to claims 38,43,49,77,89,92,95,98,101,104,107,110:**

Toy discloses a cache addressing arrangement in a segmented computer system which stores a subset of the least significant real address bits obtained by translation of a previous virtual address and uses this subset in subsequent cache accessing operations(Abstract). Toy teaches the means for generating an actual physical address from a virtual address in a time period T, by calculating a linear address based on the virtual address and calculating the actual physical address based on the calculated linear address as address translation buffer(ATB 102)(Figure 1). The ATB inherently generates a linear address and physical address based on the virtual address, which comprises segment, page, word address information(Abstract, col 1 lns 20-31, col 3 lns 58-102). He teaches a fast physical address generator for generating a fast physical address related to the virtual address in a time < T as cache address unit 125, which generates a speculative address faster than the completion of the generation of the actual physical address(Figure 1, Abstract, col 3 lns 54-col 4 lns 36).

**As to claims 39,44,50,58,62:**

The fast/speculative address can be used to access the memory faster than the actual physical address(Abstract).

**As to claims 40,45,51,59,63,67,71,79,83,84,88:**

Toy teaches determining whether the fast physical address can be referenced and canceling the memory reference if the fast physical address and the actual physical address are different(col 5 lns 13-49).

**As to claims 41,46,52,55,68,72,91,97,103,109:**

Toy teaches generating the fast physical address based on the virtual address and partial linear address information by using the virtual address and a subset of the least significant real address bits obtained by translation of a previous virtual address to generate speculative addresses(Abstract).

**As to claims 42,47,64,69,73,75,76,78,81:**

He teaches a fast physical address generator for generating a fast physical address faster than the completion of the generation of the actual linear and physical addresses(Figure 1, Abstract, col 3 lns 54-col 4 lns 36).

**As to claims 48,53,56,65:**

Toy teaches generating the fast physical address based on some of the virtual address information and partial physical address information by using the virtual address and a subset of the least significant real address bits obtained by translation of a previous virtual address to generate speculative addresses(Abstract).

**As to claim 60:**

Toy teaches that the fast physical address comprises portion of the page frame that was from a prior physical address information and inherently the offset information from the segment information of the last virtual address(Abstract, col 1 lns 30-31, col 3 ln 54-col 4 lns 36).

**As to claims 54,57,61,66,70,80,86,87:**

Toy discloses a cache addressing arrangement in a segmented computer system which stores a subset of the least significant real address bits obtained by translation of a previous virtual address and uses this subset in subsequent cache accessing operations(Abstract). Toy stores portion of a physical address of a previous translation and using that information with a virtual address to generate speculative addresses(Abstract). Toy inherently teaches generating a linear address based on a virtual address because he teaches the ATB 102 to generate the physical address based on virtual address information(Abstract, col 1 lns 20-31, col 3 lns 58-102), which inherently comprises generating an intermediate linear address. He teaches a fast physical address generator for generating a fast physical address as cache address unit 125, which generates a speculative address faster than the generation of the actual physical address(Figure 1, Abstract, col 3 lns 54-col 4 lns 36). The fast/speculative physical address can be used to reference the memory.

**As to claims 74:**

Toy discloses a cache addressing arrangement in a segmented computer system which stores a subset of the least significant real address bits obtained by translation of a previous virtual address and uses this subset in subsequent cache accessing operations(Abstract). Toy stores portion of a physical address of a previous translation and using that information with a virtual address to

generate speculative addresses(Abstract). Toy inherently teaches generating a linear address based on a virtual address because he teaches the ATB 102 to generate the physical address based on virtual address information(Abstract, col 1 lns 20-31, col 3 lns 58-102), which inherently comprises generating an intermediate linear address. He teaches a fast physical address generator for generating a fast physical address as cache address unit 125, which generates a speculative address faster than the generation of the actual physical address(Figure 1, Abstract, col 3 lns 54-col 4 lns 36). The fast physical address can be used to reference the memory. Toy teaches canceling the memory reference if the fast physical address and the actual physical address are different(col 5 lns 13-49).

**As to claims 82:**

Toy discloses a cache addressing arrangement in a segmented computer system which stores a subset of the least significant real address bits obtained by translation of a previous virtual address and uses this subset in subsequent cache accessing operations(Abstract). Toy teaches the means for generating an actual physical address from a virtual address, by calculating a linear address based on the virtual address and calculating the actual physical address based on the calculated linear address as address translation buffer(ATB 102)(Figure 1). The ATB inherently generates a linear address and physical address based on the virtual address, which comprises segment, page, word address information(Abstract, col 1 lns 20-31, col 3 lns 58-102). He teaches a fast physical address generator for generating a fast physical address related to the virtual address as cache address unit 125, which generates a speculative address faster than the completion of the

generation of the actual physical address(Figure 1, Abstract, col 3 lns 54-col 4 lns 36). The bus interface circuit for initiating a fast memory access to a memory subsystem is taught as process 101, which initiates the memory access(Figure 1).

**As to claims 85,93,99,105,111:**

Toy teaches performing an actual memory reference after the fast memory reference is canceled(col 5 lns 13-49).

**As to claims 90,96,102,108:**

Toy discloses the comparator for determining whether the second address translation can be used for a memory access as comparator 109(col 5 lns 13-49).

**As to claims 94,100,106,112:**

Toy discloses a register for storing address information of previous virtual address(col 4 lns 25-37).

***Conclusion***

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Than Nguyen whose telephone number is (703) 305-3866.

9. Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Than Nguyen
January 19, 1999

EDDIE P. CHAN
SUPERVISORY PATENT EXAMINER

☆U.S.G.P.O.: 1994-508-112

TO SEPARATE, HOLD TOP AND BOTTOM EDGES, SNAP–APART AND DISCARD CARBON

| FORM PTO-892 (REV. 2-92) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE | SERIAL NO. | GROUP ART UNIT | ATTACHMENT TO PAPER NUMBER | |
|---|---|---|---|---|
| **NOTICE OF REFERENCES CITED** | 08/905,356 | 2751 | | 8 |
| | APPLICANT(S) Belgard | | | |

**U.S. PATENT DOCUMENTS**

| * | | DOCUMENT NO. | DATE | NAME | CLASS | SUB-CLASS | FILING DATE IF APPROPRIATE |
|---|---|---|---|---|---|---|---|
| * | A | 4400774 | 2/2/81 | Toy | 711 | 3 | |
| | B | | | | | | |
| | C | | | | | | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |
| | G | | | | | | |
| | H | | | | | | |
| | I | | | | | | |
| | J | | | | | | |
| | K | | | | | | |

**FOREIGN PATENT DOCUMENTS**

| * | | DOCUMENT NO. | DATE | COUNTRY | NAME | CLASS | SUB-CLASS | PERTINENT SHTS. DWG | PERTINENT PP. SPEC. |
|---|---|---|---|---|---|---|---|---|---|
| | L | | | | | | | | |
| | M | | | | | | | | |
| | N | | | | | | | | |
| | O | | | | | | | | |
| | P | | | | | | | | |
| | Q | | | | | | | | |

**OTHER REFERENCES (Including Author, Title, Date, Pertinent Pages, Etc.)**

| | | |
|---|---|---|
| | R | |
| | S | |
| | T | |
| | U | |

| EXAMINER | DATE |
|---|---|
| T Nguyen | 1/14/98 |

* A copy of this reference is not being furnished with this office action.
(See Manual of Patent Examining Procedure, section 707.05 (a).)

RAB 97-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | |
|---|---|
| In re Application of: *Belgard, R.* | ) Art Unit: *2751* |
| | ) Examiner: *Nguyen, T* |
| Serial No.: *08/905,356* | ) |
| Filed: *8/4/97 as continuation of parent application serial no. 08/458,479 filed 6/2/95* | ) |
| For: *Fast Address Generator For Reducing Virtual-Linear-Physical Address Conversion Time* | ) |

## AMENDMENT AND RESPONSE TO OFFICE ACTION

Honorable Commissioner
of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

Applicant respectfully submits the following amendment and response to place this case in condition for allowance:

IN THE CLAIMS

**Please amend the claims as follows:**

38. (Twice amended) A system for performing address translations usable by a processor employing both segmentation and optional independent paging [said system generating an actual physical address from a virtual address in a time period T, by calculating a linear address based on said virtual address, and by calculating said actual physical address based on said calculated linear address, said system further including] the system comprising:

means for generating an actual physical address from a virtual address in a time period T, said virtual address having both a segment identifier and a segment offset by calculating a linear address based on said entire virtual address, and by calculating said actual physical address based on said calculated linear address; and

a fast physical address generator for generating a fast physical address related to said virtual address in a time < T.

43. (Amended) A system for performing address translations using a first operation to convert a first virtual address having both a segment identifier portion and a segment offset portion to a first linear address, the first linear address being based on all portions of the virtual address and a second operation to convert said first linear address to a first physical address, said system further including:

a tentative physical address generator for generating a tentative physical address related to said first virtual address;

wherein the tentative physical address can be generated before said second operation has completed converting said first linear address.

49. (Amended) A computer system which performs [for performing] address translations using a first operation to convert virtual addresses having both a segment identifier portion and a segment offset portion to linear addresses, such that both the segment identifier and segment offset portions of the virtual addresses are used for converting said linear addresses and a second operation to convert said linear addresses to physical addresses, said system further including:

a fast physical address generator for generating fast physical addresses related to said virtual addresses;

wherein the fast physical addresses can be generated while or before said virtual addresses are [being] converted in said first operation into said linear addresses.

54. (Amended) A system for performing address translations comprising:

a virtual to linear address converter circuit for generating a calculated linear address based on a virtual address, said virtual address having both a segment identifier and a segment offset, and said calculated linear address being based on all of said virtual address; and

a linear to physical address converter circuit for generating a calculated physical address based on the calculated linear address, the calculated physical address including a calculated page frame and a calculated page offset; and

a fast physical address circuit for generating a fast physical address including a fast page frame and a fast page offset;

wherein a memory reference can be generated based on the fast physical address.

57. A system for performing address translations using a first operation to convert a first virtual address <u>having both a segment identifier portion and a segment offset portion</u> to a first linear address, <u>such that all portions of the virtual address are considered when converting said virtual address into the first linear address</u> and a second operation to convert said first linear address to a first physical address, the system further including:

an address translation memory, accessible by said system while said first operation is converting said first virtual address, and capable of storing prior physical address information generated during a prior address translation by said second operation based on a prior virtual address;

wherein a fast physical address can be generated based on the prior physical address information and said first linear address before said second operation has completed converting said first linear address.

61. (Amended) A <u>computer</u> system <u>using segmentation and optional independent paging</u> for performing address translations comprising:

an address translation memory capable of storing:

(i) a portion of a physical address corresponding to a stored page frame; and

(ii) segment base information relating to a virtual address; and

a virtual to linear address converter circuit for generating a calculated linear address based on combining [a] <u>segment offset</u> portion of the virtual address and the segment base, <u>wherein all of said virtual address is used for generating the calculated linear address</u>; and

a linear to physical address converter circuit for receiving and generating a calculated physical address based on the calculated linear address, the calculated physical address including a first page frame and a first page offset; and

a fast physical address circuit for generating a fast physical address comprised of the stored page frame combined with a fast page offset portion derived from the segment base and the virtual address;

wherein the fast physical address is calculated prior to the generation of said calculated physical address.

66. (Amended) A method of performing a translation of a virtual address in a computer system using segmentation and optional independent paging, said method including the steps of :

(a) calculating a fast physical address related to said virtual address; and

(b) calculating a linear address based on said virtual address, said linear address being based on both a segment identifier and segment offset portion of said virtual address; and

(c) calculating an actual physical address based on the linear address;

wherein step (a) is completed prior to the completion of step (c), and the fast physical address can be used to initiate a fast memory reference.

70. (Amended) A method of generating memory references based on virtual addresses in a computer system, said computer system using segmentation and optional independent paging, the method including the steps of:

(a) generating tentative memory references based on said virtual addresses; and

(b) converting said virtual addresses to linear addresses during a segmentation operation, said linear addresses being based on translating all portions of said virtual address; and

(c) converting said linear addresses to physical addresses during a paging operation, so that actual memory references can be made based on said physical addresses;

wherein the tentative memory reference can be generated while said virtual addresses are being converted in said first operation into said linear addresses.

74. (Amended) A method of generating a fast memory reference using a fast physical address derived from a virtual address having both a segment identifier and a segment offset in a computer system employing both segmentation and optional independent paging, the method including the steps of:

(a) converting a portion of said virtual address into a partial linear address; and

(b) combining the partial linear address with physical address information obtained from a prior memory reference to generate said fast physical address;

(c) generating a memory reference based on the fast physical address;

(d) converting said virtual address into an actual physical address during which time a linear address is also calculated based on both the <segment id> and <segment offset> of said virtual address;

(e) cancelling the memory reference if the fast physical address and actual physical address are different.

77. (Amended) A method of generating physical addresses from virtual addresses in a computer system employing both segmentation and optional independent paging, the method including the steps of:

(a) generating a first calculated linear address based on a first virtual address in a first operation, said linear addresses being based on translating all portions of said first virtual address; and

(b) generating a fast physical address in a second operation, the fast physical address including linear address information relating to said first virtual address and portions of physical address information relating to said first virtual address; and

(c) generating a first calculated physical address in a third operation based on the first calculated linear address;

wherein the fast physical address is generated prior to the generation of the first calculated physical address.

82. (Amended) A system for performing memory references in a processor which employs both segmentation and optional <u>independent</u> paging during an address translation, said system comprising:

means for performing an address translation by generating a first physical address from a first virtual address by first calculating a first linear address based on <u>both</u> a first segment identifier and first offset associated with the first virtual address, <u>such that all of said first virtual address is translated</u>, and then calculating the first physical address based on the first calculated linear address; and

a fast physical memory access circuit for generating a fast memory reference, which fast memory reference is based on physical address information from said address translation means;

a bus interface circuit for initiating a fast memory access to a memory subsystem based on said fast memory reference.

86. (Amended) A method for performing memory accesses between a processor and a memory, said processor having an address translation mechanism that employs segmentation and optional <u>independent</u> paging, the method comprising the steps of:

generating computed physical addresses by converting virtual addresses having a segment identifier and a segment offset into linear addresses, <u>such that all portions of said virtual addresses are translated</u>, and then converting said linear addresses into a physical addresses;

generating a speculative physical address based on one of said computed physical addresses;

initiating a speculative memory access based on said speculative physical address.

89. (Amended) A system for performing a first and a second address translation of first and second virtual addresses respectively, the system comprising:

a virtual to linear address converter circuit for generating a first calculated linear address based on translating all portions of the first virtual address including a segment identifier and a segment offset; and

a linear to physical address converter circuit for completing the first address translation by generating a first calculated physical address based on said first calculated linear address, said first calculated physical address including a first calculated page frame and a first calculated page offset; and

wherein the system uses information from the first address translation during the second address translation so that the second address translation can be performed faster than the first address translation.

95. (Amended) A circuit for performing fast translations of virtual addresses to physical addresses in a computer system which uses both segmentation and optional independent paging, the circuit including:

an address generator for performing a first address translation of a first virtual address having an associated first segment identifier and a first offset, said first translation including converting all of said virtual address into a first linear address;

said address generator also performing a fast address translation of a second virtual address having an associated second segment identifier and a second offset, said fast address translation occurring without converting all of said second virtual address into a second linear address;

wherein said address generator uses information from the first address translation during the fast address translation so that said translation of said second virtual address takes less time than said first address translation.

101. (Amended) A method of translating virtual addresses in a computer system that uses both segmentation and optional independent paging, the method including the steps of:

(a) generating a first calculated physical address based on a first virtual address in a first operation, said first virtual address including a first segment identifier and a first offset and wherein said first operation converts all of said virtual address into a first linear address; and

(b) generating a second fast physical address in a second operation based on a second virtual address, said second virtual address including a second segment identifier and a second offset, and said second fast physical address being generated based on information obtained during said first operation, and without converting all of said second virtual address into a second linear address;

wherein said second operation is performed faster than said first operation.

107. (Amended) A method of performing address translations in a computer systemthat uses both segmentation and optional independent paging, the method including the steps of:

(a) performing a first address translation by translating a first virtual address into a first physical address by: (i) first calculating a first linear address based on a first segment identifier and first offset associated with said first virtual address wherein all of said virtual address is translated; and (ii) calculating said first physical address based on said first calculated linear address and

(b) performing a second address translation using information obtained during said first address translation to translate a second virtual address into a second physical address, said second physical address being obtained without converting all of said second virtual address into a second linear address;

wherein said second translation can be achieved in less time than said first translation.

Remarks

Claims 38 – 112 are pending. Independent claims 38, 43, 49, 54, 57, 61, 66, 70, 74, 77, 82, 86, 89, 95, 101 and 107 are amended. The following comments are provided in response to the Examiner's rejections of the claims in light of Toy under § 102 in ¶7 of the Office Action.

Initially, Applicant notes that there seems to be some confusion about what kind of address translation is shown in Toy (U.S. Patent No. 4,400,774). The thrust of the Toy teachings, in fact, are only concerned with data access from a cache *subsequent* to an actual virtual-to-physical translation. There is very little detail, if any, on the actual operation of the address translation buffer.

Apparently, however, there is a perception by the Examiner that this reference depicts a virtual-linear-physical address system, and on this basis the Examiner believes there is "inherently" a linear address generated. *See e.g.*, Office Action, page 3, line 9. This is incorrect, nonetheless, and this can be seen quite plainly from the discussion that follows.

Because the Examiner's rejection based on this reference appears to be based on a faulty assumption, and because there is no prior art known to Applicant that describes or suggests the limitations of the pending claims, Applicant submits that the claims are in condition for allowance.

To begin with, Toy is typical of prior art systems where segmentation and paging are *integrated*, and hence transformation from virtual to physical addresses is performed in one step; *i.e.*, there are no separate segmentation/paging operations during an address translation. This can be verified from the fact that the virtual address in Toy is specifically noted as having a format of the type [segment:page:word]. Toy specifically indicates that the word address bits *are not even translated. See e.g.*, c.3 64 - 69. From this fact alone it can be seen that it is not a true separated segmentation/paging system as in the present invention, *and thus there never is any "linear" address of any kind.*

One of the objects of the present invention as disclosed in the specification on page 4 at lines 22-26, is to achieve the speed and performance advantages of *integral* segmentation and paging, and at the same time, to provide the space compaction and compatibility advantages of *separate* segmentation and paging. Integral segmentation and paging systems such as Toy cannot provide this capability, and do not suggest to - let alone instruct - one of skill in the art how to do so. Moreover, such systems have a number of drawbacks, including the fact that the segments must start on page boundaries, and must be comprised of an integral number of pages, regardless of size.

The present invention is used in environments that have virtual addresses composed of a segment identifier and a segment offset; for a normally calculated physical address, these virtual

addresses must be converted by a segmentation process into a linear address. During the virtual-to-linear translation, the segment offset portion (the lower portion of the virtual address) contributes to both the page and word (or byte) offset of the linear address. This is characteristic of the two step (virtual-to-linear, then linear-to-real) independent segmentation and paging scheme of the environments where the present invention is used. Thus, there is no such separate segmentation process or apparatus disclosed or taught by <u>Toy</u>, because there is no such linear address.

Accordingly, <u>Toy</u> is quite easily distinguishable from the present claims because it does not teach or disclose: (1) a virtual address of the [segment identifier: segment offset] variety; or (2) a linear address of the type used in the present invention; or (3) a virtual-linear address conversion operation. Again, applicant believes the claims <u>already</u> differentiate from such a system, but have further amended the claims to better clarify the scope of the present invention. As the Examiner can verify, the new amendments track these distinctions fairly closely. To further distinguish claims 81 – 112, they have been amended to indicate that, unlike <u>Toy</u>, <u>all of the virtual address is translated</u> when a calculated linear address is required.

In brief, Applicant was the first to conceive and describe a viable mechanism that provides fast memory references in computer systems that use independent segmentation plus optional paging for address translations. This type of speculative operation allows for faster program execution, because data at a particular physical memory location can be retrieved faster than with conventional prior art memory access schemes. There is simply no teaching, suggestion or hint in <u>Toy</u> (or elsewhere) on how to achieve such functionality in the kind of environment where the applicant's invention is used. Accordingly, applicants submit that the claims should be in condition for allowance.

Should the Examiner believe it that it would be helpful to discuss any of the above points in person, Applicant is open to a telephone conference (408-342-1862) at any convenient time.

Respectfully submitted,

Date: July 30, 1999

J. Nicholas Gross, Attorney, Reg. No. 34,175

*I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Commissioner of Patents and Trademarks, this 30th day of July 1999.*

MAR 2 2 2000

O I P E PATENT & TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

AY

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. |
|---|---|---|---|
| 08/905,356 | 08/04/97 | BELGARD R | RAB-97-002 |

LM02/1026

PETER COURTURE
LAW+
993 HIGHLAND CIRCLE
LOS ALTOS CA 94024

| EXAMINER | |
|---|---|
| NGUYEN, T | |
| ART UNIT | PAPER NUMBER |
| 2751 | 12 |

DATE MAILED: 10/26/99

**Please find below and/or attached an Office communication concerning this application or proceeding.**

**Commissioner of Patents and Trademarks**

PTO-90C (Rev. 2/95)

2- Mail Copy

| # Office Action Summary | Application No. 08/905,356 | Applicant(s) Belgard |
|---|---|---|
| | Examiner T Nguyen | Group Art Unit 2757 |

**—The MAILING DATE of this communication appears on the cover sheet beneath the correspondence address—**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, such period shall, by default, expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).

**Status**

☒ Responsive to communication(s) filed on 1/19/99 + 8/2/99.

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle,* 1935 C.D. 11; 453 O.G. 213.

**Disposition of Claims**

☒ Claim(s) 38-112 is/are pending in the application.

Of the above claim(s) ______ is/are withdrawn from consideration.

☐ Claim(s) ______ is/are allowed.

☒ Claim(s) 38-112 is/are rejected.

☐ Claim(s) ______ is/are objected to.

☐ Claim(s) ______ are subject to restriction or election requirement.

**Application Papers**

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The proposed drawing correction, filed on ______ is ☐ approved ☐ disapproved.

☐ The drawing(s) filed on ______ is/are objected to by the Examiner.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. § 119 (a)-(d)**

☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) ______.

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: ______.

**Attachment(s)**

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). ______

☐ Interview Summary, PTO-413

☒ Notice of Reference(s) Cited, PTO-892

☐ Notice of Informal Patent Application, PTO-152

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Other ______

**Office Action Summary**

## DETAILED ACTION

1. This is a response to the declaration, filed 1/19/99, and amendment, filed 8/2/97.

2. Claims 38-112 are pending.

### *Response to Arguments*

3. Applicant's arguments with respect to claims 38-112 have been considered but are moot in view of the new ground(s) of rejection. Applicant's amendment to the claims requires new search and consideration. The amended claims are rejected using new arts. Accordingly, this office action is made final.

### *Claim Rejections - 35 U.S.C. § 103*

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 38-112 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crawford (US 5,321,836) in view of Toy (US 4,400,774).

**As to claims 38,43,48-49,53,54,56-57,60-61,66,70,77,82,86,87,89,92,95,98,101,104,107,110:**

Crawford discloses a virtual memory management method and apparatus using segmentation and optional, independent paging mechanism. Crawford teaches the means for generating an actual physical address from a virtual address in a time period T (Figures 2 & 3, physical address is generated from a virtual address). Crawford teaches the virtual address

having both a segment identifier and a segment offset (Figure 2, virtual address has segment id and offset). Crawford teaches calculating a linear address based on the entire virtual address (linear address 32 (Figure 3) is calculated through segmentation, using the entire virtual address (Figure 2). Crawford teaches calculating the actual physical address based on the calculated linear address (the actual physical address is calculated from the linear address including a page frame and page offset through paging, Figure 3). Crawford teaches a bus interface circuit for physical memory access (Figure 1, interface between bus unit and main memory 13).

Crawford does not specifically teach generating a fast physical address related to the virtual address in a time < T. It is well-known in the art to generate fast/speculative/prefetch/predictive addresses to obtain faster access. For example, Toy discloses a non-segmented memory system with a speculative address generator as cache address unit 125, which uses previous address bits to predict new physical addresses(col 3 ln 54 - col 4 ln 25, Figure 1). This provides for a faster physical address generation if the desired physical address shares the same address bits. The speculative physical address can be generated before the actual/normal linear address is calculated (col 3 ln 54 - col 4 ln 25). The speculative address can be used for generating a faster memory access. Accordingly, it would have been obvious to one of ordinary skills in the art at the time of the invention to use Toy's teachings to generate a fast/speculative/predictive physical address related to the virtual address in a time < T in Crawford's segmented memory system so that memory access can be performed more quickly.

**As to claims 39,44,50,58,62,78,81:**

Crawford in view of Toy teaches the fast/speculative/predictive physical address being generated for a faster memory access than through using the actual physical address, which is generated after the fast/speculative physical address (See response to claim 38).

**As to claims 40,45,51,59,63,67,71,74,79,83,84,88:**

Toy teaches canceling the memory access if the fast physical address does not equal the actual physical address(col 5 lns 13-49), otherwise the fast physical address access is completed.

**As to claims 41,46,52,55,68,72,80,91,97,103,109:**

Crawford in view of Toy teaches the fast physical address is generated based on combination of physical address information from a different virtual address, and partial linear address information relating to the virtual address by using previous address bits to generate the fast physical address(Toy col 4 lns 1-36; Crawford Figures 2 & 3).

**As to claims 42,47,64,69,73,75,76:**

Crawford in view of Toy teaches that the fast physical address can be generated before the actual/normal linear address is calculated (col 3 ln 54 - col 4 ln 25).

**As to claim 65:**

Toy teaches generating the speculative address using part of the previous address(col 3 ln 54 - col 4 ln 25).

**As to claims 85,93,99,105,111:**

Toy also teaches performing an actual memory reference after the fast memory reference is canceled(col 5 lns 13-49).

**As to claims 90,96,102,108:**

Toy also discloses the comparator for determining whether the second address translation can be used for a memory access as comparator 109(col 5 lns 13-49).

**As to claims 94,100,106,112:**

Toy also discloses a register for storing address information of previous virtual address(col 4 lns 25-37).

***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Than Nguyen whose telephone number is (703) 305-3866.

7. Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Than Nguyen
October 21, 1999

Eddie C

EDDIE P. CHAN
SUPERVISORY PATENT EXAMINER

FORM PTO-892

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE



# NOTICE OF REFERENCES CITED

| SERIAL NO. | GROUP ART UNIT | ATTACHMENT TO PAPER NO. |
|---|---|---|
| 08/905,356 | 2751 | 12 |

APPLICANT(S): Belgard

## U.S. PATENT DOCUMENTS

| * | | DOCUMENT NO. | DATE | NAME | CLASS | SUB-CLASS | FILING DATE |
|---|---|---|---|---|---|---|---|
| | A | 5,321,836 | 6/1994 | Crawford et al. | 711 | 206 | |
| * | B | 4,400,774 | 8/1983 | Toy | 711 | 3 | |
| | C | | | | | | |
| | D | | | | | | |
| | E | | | | | | |
| | F | | | | | | |
| | G | | | | | | |
| | H | | | | | | |
| | I | | | | | | |
| | J | | | | | | |
| | K | | | | | | |

## FOREIGN PATENT DOCUMENTS

| * | | DOCUMENT NO. | DATE | COUNTRY | NAME | CLASS | SUB-CLASS |
|---|---|---|---|---|---|---|---|
| | L | | | | | | |
| | M | | | | | | |
| | N | | | | | | |
| | O | | | | | | |
| | P | | | | | | |
| | Q | | | | | | |

## OTHER REFERENCES (Including Author, Title, Date, Pertinent Pages, Etc.)

| | |
|---|---|
| R | |
| S | |
| T | |
| U | |

| EXAMINER | DATE | |
|---|---|---|
| Than Nguyen | October 21, 1999 | Form892ccs2106b |

* A copy of this reference is not being furnished with this office action.
(See Manual of Patent Examining Procedure, section 707.05(a).)